

CSE 240 Spring 2020 Homework 6, Linked Lists (50 points)

Due Saturday, February 29, 2020 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. You are given a partially completed program hw06q1.c. The structure of this homework is similar to previous homework. In this homework, you should use linked list to do the same work in the previous homework.

You should follow the instructions given in the program to complete the functions so that the program executes properly. You will be completing a program that creates a linked list of books.

It is a menu-driven program where the user is given the following options:

- a) Add a new student to the list. When adding a new student to the list, the user is prompted for student's name, major, school year of student and ID number of the student. The student should be added at sorted position. If the student already exists in the list, then you should not add to the list. The school year is enum type. The sorting should happen within the list. You should not create a new linked list of students having sorted students.
- b) Display the list of students. This should display each student's details one after the other.
- c) Remove the student from the linked list. After removing the student, the linked list should not be broken.

Expected output of each menu option: (similar as previous homework)

addSort:

```
CSE240 HW6

Currently 0 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
a

Enter student name: Eden Hazard
Enter major: CSE
Enter whether student is 'freshman' or 'sophomore' or 'junior' or 'senior': junior
Please enter ID number: 111

student successfully added to the list!

Currently 1 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
a

Enter student name: William Borges
Enter major: EE
Enter whether student is 'freshman' or 'sophomore' or 'junior' or 'senior': senior
Please enter ID number: 222

student successfully added to the list!
```

```
Currently 2 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
a

Enter student name: David Luiz
Enter major: CSE
Enter whether student is 'freshman' or 'sophomore' or 'junior' or 'senior': freshman
Please enter ID number: 333

student successfully added to the list!
```

displayList:

```
Currently 3 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
d

Student name: Eden Hazard
Student major: CSE
SchoolYear: junior
ID number: 111

Student name: David Luiz
Student major: CSE
SchoolYear: freshman
ID number: 333

Student name: William Borges
Student major: CSE
SchoolYear: senior
ID number: 222

Currently 3 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
```

deleteNode: (see displayList() photo for list before deleting node)

```
Currently 3 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
r

Please enter student name: Eden Hazard

Student successfully removed from the list.

Currently 2 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
d

Student name: David Luiz
Student major: CSE
SchoolYear: freshman
ID number: 333

Student name: William Borges
Student major: CSE
SchoolYear: senior
ID number: 222

Currently 2 student(s) on the list.
Enter your selection:
    a: add a new student
    d: display student list
    r: remove a student from the list
    q: quit
```

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.