

CSE 240 Spring 2021 Homework 5: Array of Structs and Enum Types (50 points)

Due Saturday, February 20, 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 basic C syntax, using structures and enumeration types. By the end of the assignment, you should have

- understood the concepts and operations of enumeration type, arrays, structures, and array of structures, and files.
- written a program using enumeration type and an array of structures.

Reading: Textbook Chapter 2, sections 2.5, and 2.6, and lecture slides covered.

Exercising: Complete the multiple choice questions in Textbook Section 2.10. The answers of the questions are available in course Web page.

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 or code in the course discussion board. **Do not cooperate with your peers in doing the individual assignments.**

Programming Assignment (50 points)

1. You are given a partially completed program hw05q1.c. You should follow the instructions given in the program to complete the functions so that the program executes as instructed. The program declares 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 (array of structures). 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) Sort the list of books numerically by ID. The sorting should happen within the list. You should not create a new list (array of structs) of books having sorted books.

- c) 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 array 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.

There is save() already implemented to write the books list to a file 'Book_List.txt'. save() is executed at the end of the program when the user quits the program. 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 array 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:

```
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 (given):

(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:

```
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:

```
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:

```
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.

What to Submit?

You are required to submit your solution in a compressed format (.zip). Make sure your compressed file is label correctly - lastname_firstname5.zip. (All lowercase, do not put anything else in the name like "hw5".)

The compressed file MUST contain the following:
hw05q1.c (completed code)

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 may consist of multiple files. You must copy these files into a single folder for blackboard 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 blackboard. 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 blackboard. 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 blackboard 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;
- 10% grade deduction for every day it is late after the grace period;
- No late submission after Tuesday at 11:59PM.

Grading 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.
- (3) 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.
- (4) 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!
- (5) We will apply the following rubrics to **each sub-question** listed in the assignment. Assume that points assigned to a sub-question is pts:

Major	Code passed compilation				Code failed compilation		
Points	Pts * 100%	Pts * 90%	Pts * 80%	Pts * 60%-70%	Pts * 40%-50%	Pts * 10%-30%	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

Please read the FAQ file in the Course Information folder:

Q: For some reason, my assignment submission did not go through, but I thought it went through. I can show you on my local disk or in my Dropbox that I completed the assignment before the due date. Can my assignment be graded?

A: You should always download your own submission from the blackboard after submission and test if the submission contains all the required files. We will grade the assignment submitted to Canvas **only**. We cannot grade the assignment sent from email or stored in any other places, regardless its last-modified-time. If you submitted your assignment into the blackboard, it cannot

be downloaded from the instructor side, but it can download from your side, we can download from your blackboard and grade the assignment. Please meet the instructor or TA in this case.