CSE 240 Spring 2021 Homework 7:
Linked List of Containers (50 points)

Due Saturday, **March 13, 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 linked list, pointer operations, and parameter passing mechanisms. This two-assignment combination allows you to put all together that you have learned in C programming language into a large program. By the end of the assignment, you should have understood and exercised

- Pointer and pointer to pointer operations accessing structured data
- Linked list of structures, with complex manipulations of pointer and structure
- Different types of parameter passing mechanisms and return values of different types.
- Solving problems in recursion

Reading: Textbook Chapter 2, Section 2.5.4 on linked list and Section 2.6 on parameter passing and Section 2.10.

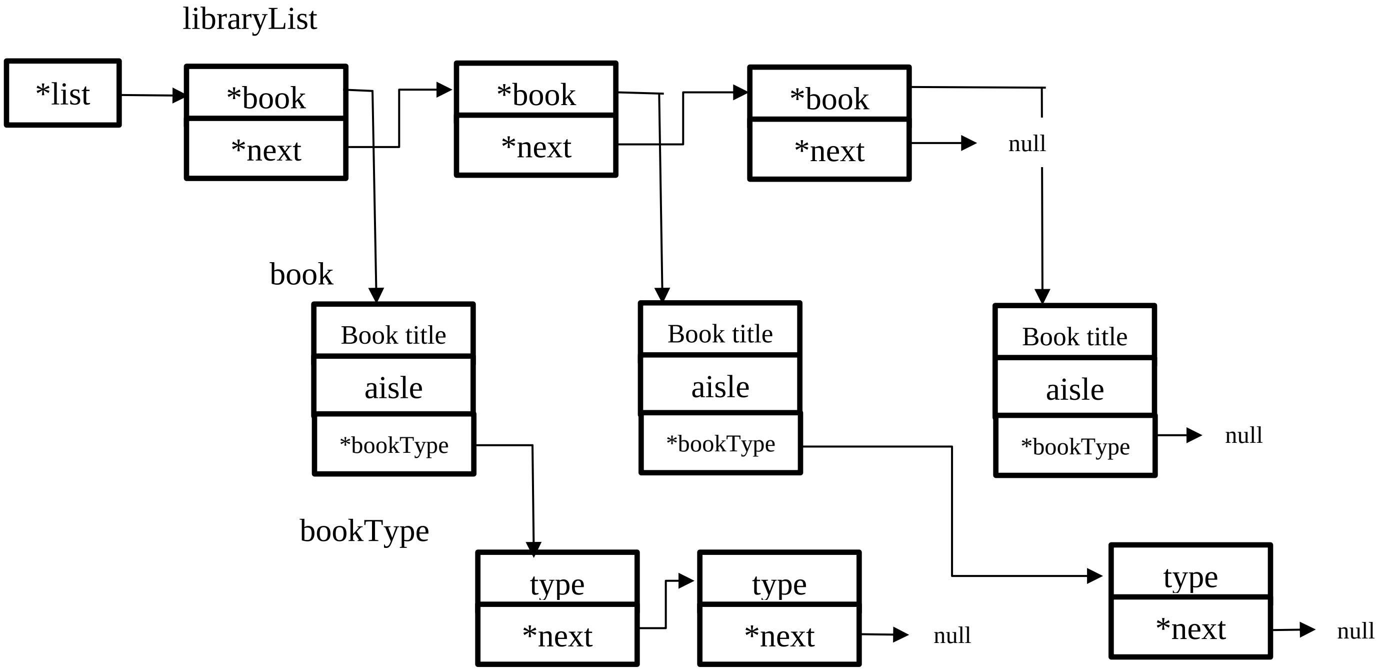
Preparation: Complete the multiple choice questions in the textbook exercise section. The answer keys can be found in the course Web site. These exercises can help you prepare for your weekly quiz and the exam. You are encouraged to read the other exercise questions and make sure you understand these 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 of class meetings. Should you need assistance, or have questions about the assignment, please contact the instructor or 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.

Homework Description

The following figure shows an instance of the linked list that you will be building and managing in these two homeworks. The node of the main linked list 'list' is called libraryList, which contains two members: (1) book member is a pointer to the book node, (2) a pointer to next libraryList node. The book node contains three members: (1) book title, (2) aisle number, and (3) a pointer to bookType list. The node in the bookType list has two members: (1) book type (2) a pointer to next bookType node.



HW 7 Programming Assignment (50 points)

You are given a partially completed program hw7.c. 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 list of books. It is a menu driven program where user is given following options:

a) Add an book’s information (Book title and aisle number) is already implemented. The new book is added to the head of the linked list. We do not ask user for bookTypes to add in this function. So simply NULL is assigned to *bookType member of the ‘book‘ node when a new book is added to the list in this function. (Note: *bookType is used in further functions)

b) Display the book list is already implemented. This function prints each book’s title and aisle number. It does not print bookType of the book.

These functions need to be implemented:

c) Search a book in the list by book title. It prints if the book exists on the list or not. This function should return the ‘book‘ node if that book is found in the list, else return NULL. It is used as a helper function in executeAction() to check if the book exists in the list.

The next part focuses on using ‘bookType’ linked list. In this part, the user should be able to use the following menu options:

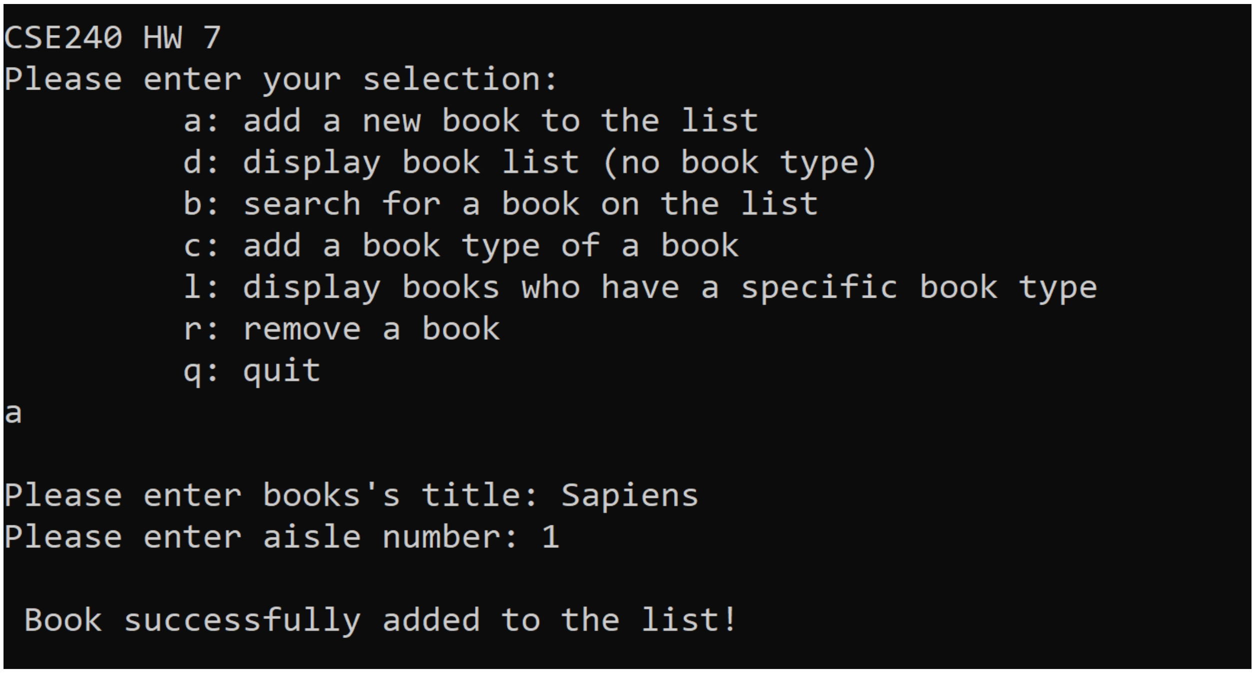
a) Add a bookType to an book’s profile. This function assumes that the book is added in the list and assigns book type using the *bookType member of ‘book’ node. You may add the new bookType to the head or tail of the ‘bookType’ linked list. (Sample solution adds the bookType to the tail)

b) This function prompts the user to input a book type. Then it displays the list of books that have the entered book type only. Lastly, it should display the number of books that met this requirement. This function should display both book's title and aisle number. See expected output below.

c) Removes the book from the list. This function removes the book's title, aisle number and bookType list of the book when removing the book from the list.

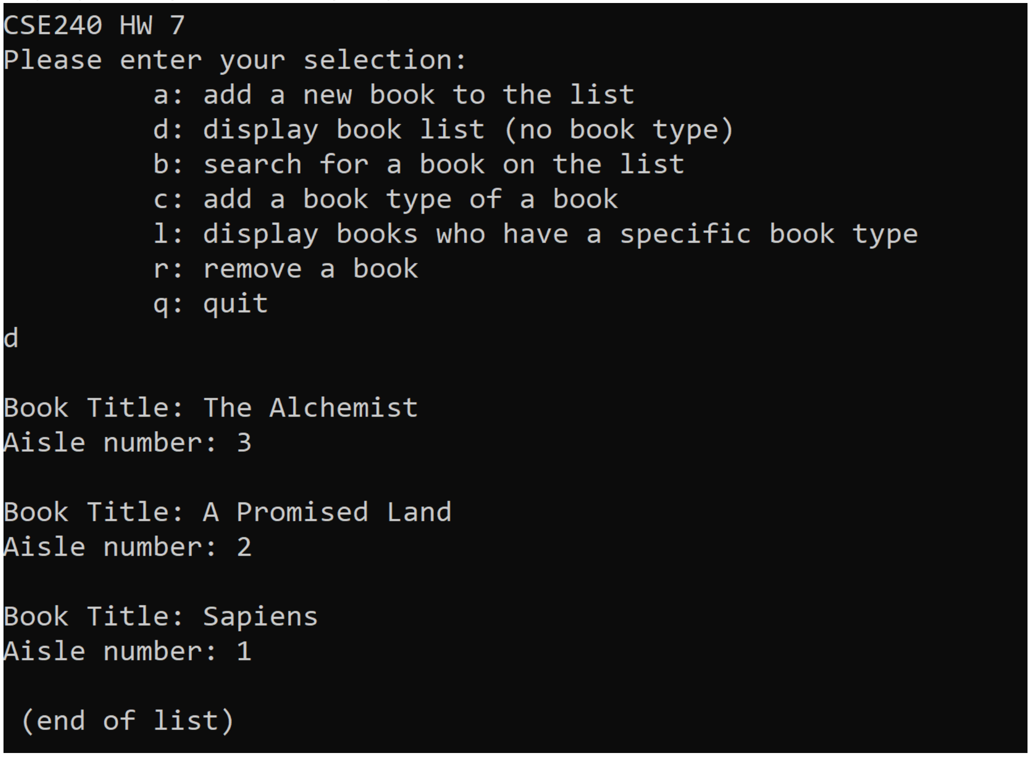
Expected outputs:

addBook (already implemented)



```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
a

Please enter books's title: Sapiens
Please enter aisle number: 1

 Book successfully added to the list!
```

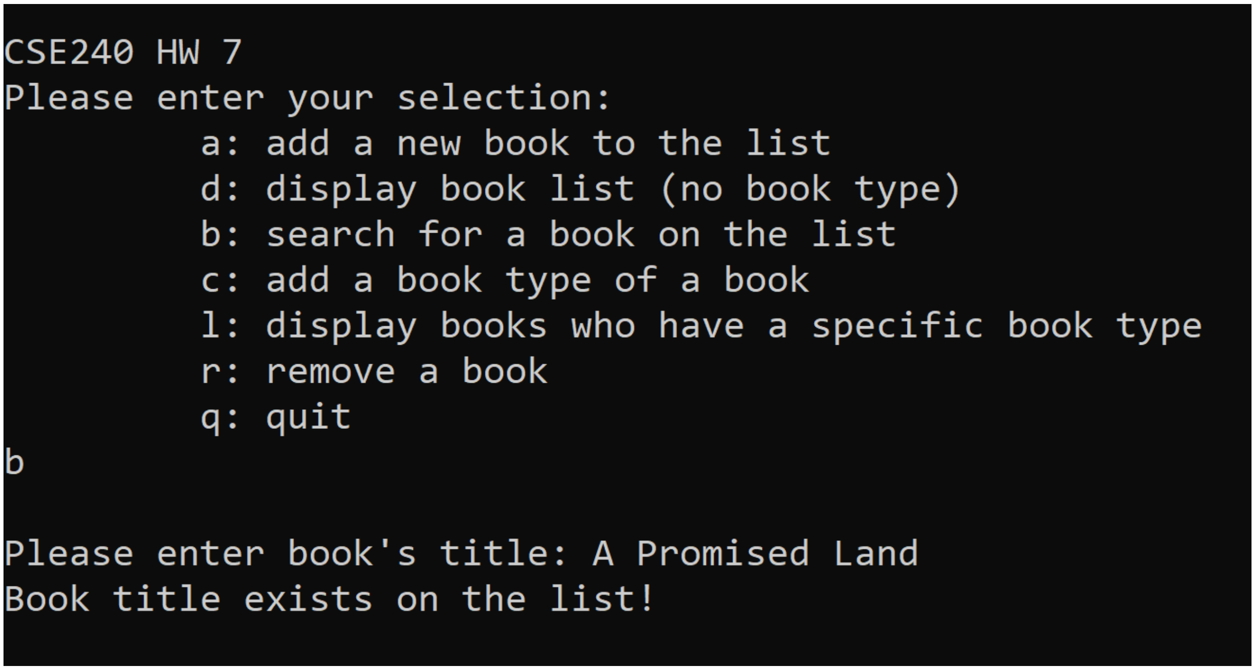
displayLibraryList (already implemenetd)



```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
d

Book Title: The Alchemist
Aisle number: 3

Book Title: A Promised Land
Aisle number: 2

Book Title: Sapiens
Aisle number: 1

 (end of list)
```

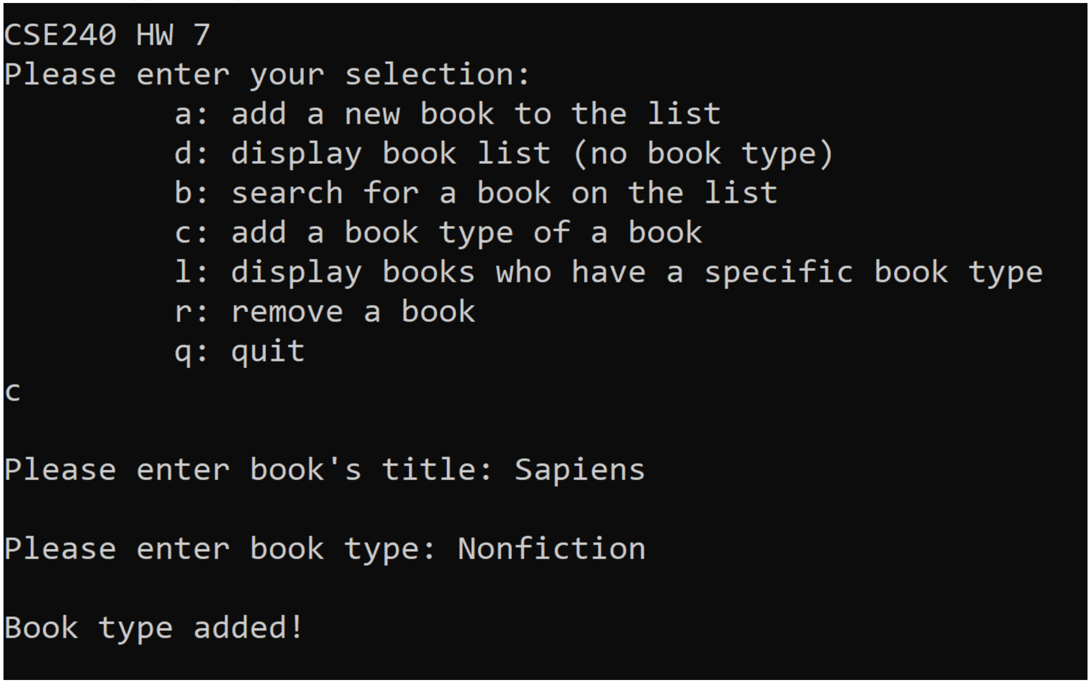
(New books are added to the head of ‘list’. So newer books appear first. Not necessary to print ‘end of list’ message.)

searchBook (5 points)



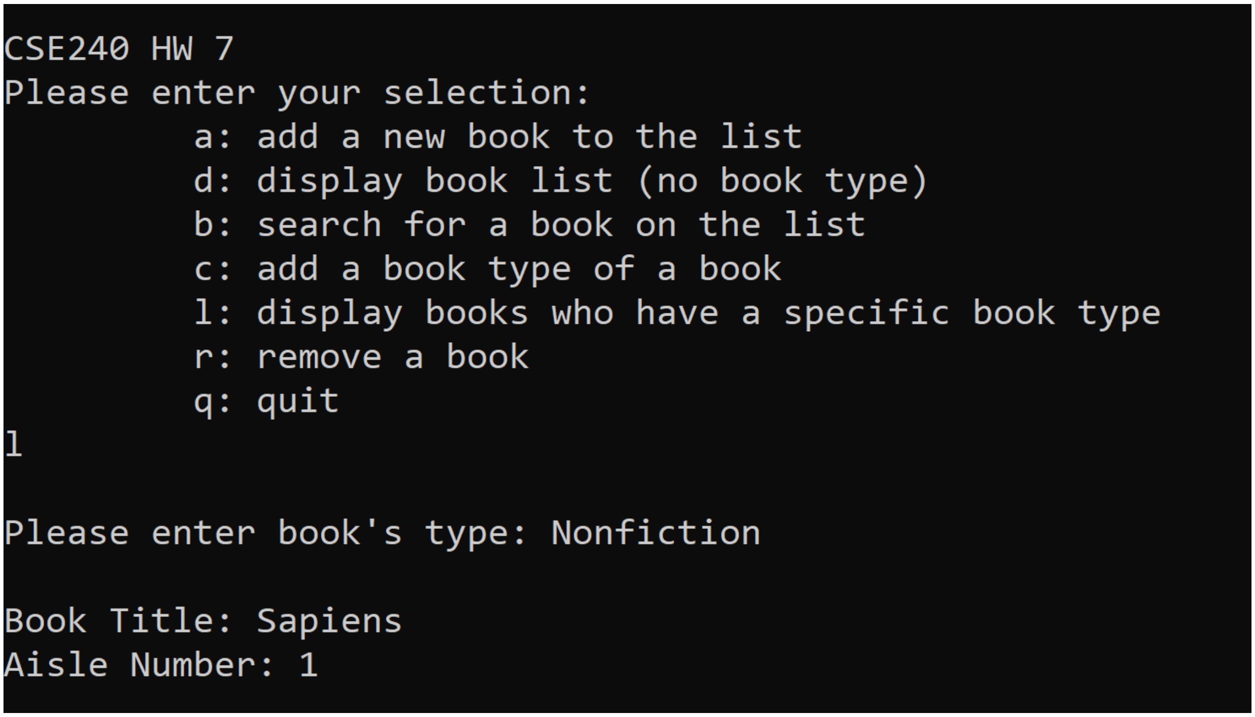
```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
b

Please enter book's title: A Promised Land
Book title exists on the list!
```

addBookType (15 points)



```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
c

Please enter book's title: Sapiens

Please enter book type: Nonfiction

Book type added!
```

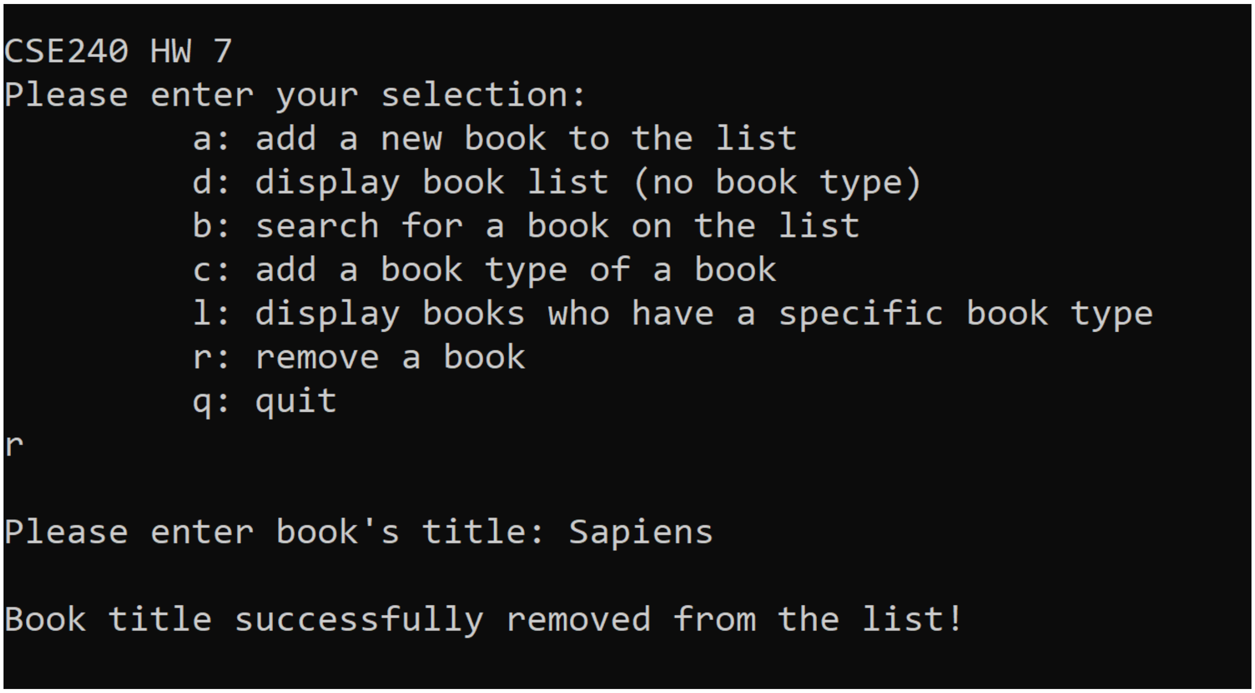
displayBookTypeList (15 points)



```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
l

Please enter book's type: Nonfiction

Book Title: Sapiens
Aisle Number: 1
```

(Sapiens recently added book type as ‘Nonfiction’ which was added to tail of her *bookType list. So, it’s name appears when ‘l’ option is used).

removeBook (15 points)



```
CSE240 HW 7
Please enter your selection:
         a: add a new book to the list
         d: display book list (no book type)
         b: search for a book on the list
         c: add a book type of a book
         l: display books who have a specific book type
         r: remove a book
         q: quit
r

Please enter book's title: Sapiens

Book title successfully removed from the list!
```

(After removing a book, you should use the display option to verify it functioned correctly)

What to Submit?

You are required to submit your solution in a compressed format (.zip). Make sure your compressed file is label correctly - lastname_firstname7.zip. (All lowercase, do not put anything else in the name like "hw7".)

The compressed file MUST contain the following:

hw7.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 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;
- 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.