Programming Assignment 3

Runhan Mao CSCE 221 section-201

07/28/16

Program Description

Part 1 of this assignment requires implementing a simple Doubly Linked List, a Doubly Linked List, and a template Doubly Linked List. Part 2 of the assignment requires the students to use the template Doubly Linked List from part 1 and write a library management system that store books. Each book stores the following information: book title, author's name, 13-digit ISBN, publishing year, and edition number. The students will design a nice interface where the user can start searching for a book by inputing the title of the book. If the book is found, the library will simply display the book. If the book is not found, the user will be ask to input information about the book and add it to the library. If more than one book has the same title and author, the user will be prompted to input the specific edition to extract the book. The program will display the book requested by the user.

Purpose of the Assignment

Part one of this assignment is to let students learn how to write data structue and implementations for DoublyLinkedList class, SimpleDoublyLinkedList class, and to convert the DoublyLinkedList class into templateDoublyLinkedList class. Later on, for part two of the assignment, the students are required to learn how to implement DoublyLinkedList class in a real world problem, in this case, implementing a library management system.

Data Structures Description

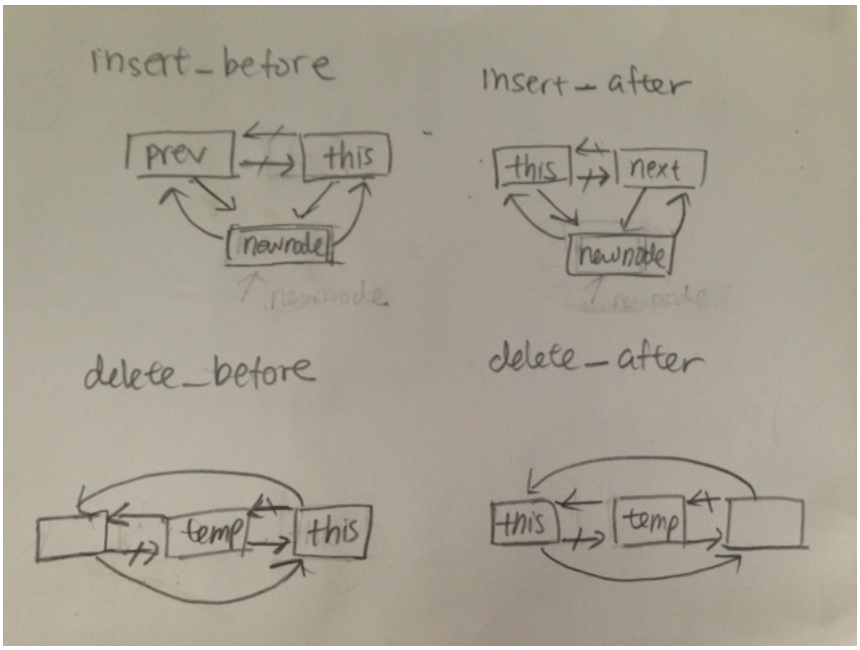
By doing this assignment, I learned about Doubly Linked List, its data structure and its implementation. Doubly Linked List has nodes that store two pointers, a pointer points to previous and a pointer points to next. It also has two sentinel nodes called header and trailer that do not store any element. Doubly Linked List has several functions, such as insertFirst(), insertLast(), removeFirst(), removeLast(), getFirst(), getAfterLast(), isEmpty(), int First(), and int Last(). In addtion, it has constructor, copy constructor, copy assignment, and

destructor. I also learned how to implement 4 major functions: insert_before, insert_after, delete_before and delete_after for SimpleDoublyLinkedList. For part two of the assignment, several functions are required for the Record class, overloading of the input operator to read in the records, overloading the output operator to print out the records, and overloading the comparison operator to compare items in the Record. In order to implement the library management systems, several functions are written. Void read() function reads from the file and store the information inside a vector consists of 26 books. Each element of the vector is a doubly linked list, and each node of the doubly linked list stores a record. Void showLibrary() displays the books in the library. Void addToLibrary() adds the book to the library if the book does not already exit in the library. Void Search() function search for user request and display the found book. Finally, void userinterface() ask the user to input the title of the book. It will ask if the user want to keep searching for another book, and it will terminates the probram if the user says no.

Algorithm Description

In part 1, for SimpleDoublyLinkedlist, we are required to implement 4 functions: insert_before, insert_after, delete_before, and delete_after. For insert_before function, a new node is allocated dynamically, and it is inserted before the "this" node, which is the node that you want to insert in front of. NewNode is initialized with element d, and its previous pointer points to previous, and its next pointer points to this. It takes constant time to insert a new node, so its running time is O(1). Its specifc running time function is shown in the code, please refer to the code for running time function. Insert_after is very similar to insert_before. Everytime a function is called, a new node is allocated dynamically. The new node's element is intialized with element d, its previous pointer points to this, and its next pointer points to next. It takes constant time to insert, so its complexity is O(1) as well. delete_before function assign a temporary node called temp before the "this" node. Its running time is O(1). delete_after assigns a temporary node called temp after the "this" node. Its running time is also O(1). I drew four pictures to illustrate the implementations for these 4 functions, it's more clearer than describing in words. For DoublyLinkedList,we are required to write 3 functions: copy constructor, copy assignment, and overloading ostream operator. copy constructor always initializes first in case of an empty linked list. It dynamically allocates a pointer that first points to dll.getFirst(), and then it iterates throught the entire linked list and copy each element until it gets to dll.getAfterLast(). Each iteration is incremented by newNode-getNext(). Copy constructor creates and initializes an object on the spot. The running time for copy constructor is O(n), since it copies n elements. Copy assignment utilizes the same algorithm, the difference is that it first delete the pre-initiazlied linked list pointed by "this", then it copies from the another linked list, and finally it returns the "this" pointer. It takes O(n) times to copy each item. Ostream operator works in a similar

way, it goes through the while loop so it can output each node of an linked list. To copy an entire linked list with n nodes, it needs to iterates n times, so its running time is O(n). For part two, inside Record.h, we have to overload the input operator to read an entire record with 5 items: title, name of the author, ISBN, year, and edtion. The function getline is used to read in empty space. set functions are use to set the variables read from the file. The running time is O(1), because it takes constant time to read 1 object with 5 items each time. The output operator prints out an entire record with the 5 items. It also takes constant time to print 1 record, so its running time is O(1). Finally, the less than operator compares title, author, and ISBN. It will first compare titles. If titles are the same, it will compare authors, and if both titles and authors are the same, it will compare ISBN. It does three comparisons in the worst case, so its running time is constant, O(1). The insertOrderly function inserts a record before the node that it is being compared, if the record is smaller. If the record is greater than all the other records in the list, it will be placed at the end of the linked list and pointed by trailer. It will take n comparisons in the worst case, so its running time is O(n). To search for an item, the funciton takes O(n) times in the worst case, and O(1) in the best case.



Program Organization and Description of Classes

In part one, there are DoublyLinkedList class, SimpleDoublyLinkedList class, and TemplateDoublyLinkedList class. There is no relation between the DoublyLinkedList class and SimpleDoublyLinkedList class. However, DoublyLinkedList class is used to convert to TemplateDoublyLinked class. Part two requires the Record class, the STL vector class and the TemplateDoublyLinkedList class. The vector class is provided by the standard library. Each vector element is a type DoublyLinkedList and each DoublyLinkedList is a type Record. The reason to use TemplateDoublyLinkedList class instead of just DoublyLinkedList class is that the DoublyLinkedList class must be generic so it can hold data of different types. In this case, it holds the type Record.

Instructions to Compile and Run your Program

For better Organization, I have four folders. One is called DoublyLinkedList, one is called SimpleDoublyLinkedList, one is called TemplateDoublyLinkedList, and finally the last one is called book, which consists of part two of this asssignment. Go to each folder/directory separately, and compile. Inside the folder DoublyLinkedList, compile main.cpp. Inside SimpleDoublyLinkedList folder, compile SimpleDoublyLinkedList.cpp. Inside TemplateDoublyLinkedList folder, compile TemplateMain.cpp. Finally, inside the book folder, compile main.cpp. To complile the program, use the linux machine command line g++ -std=c++11 *.cpp or make all. To run the program, execute ./main. Before executing ./main, make sure a makefile has already been created first.

Input and Output Specifications

There are no specific input specifications for part 1 of this assignment, since there is no input requried for testing. For part two, there are several specifications for user input in order to run the program smoothly. First, when user is requested to input the title of the book, the first letter of the title has to be capital. If user accidentally typed lowercase for the first letter of the title, the program will throw an out of range error. Second, the user should input the title exactly as shown in the library. If the user mispelled or mistyped the title of the book, the program will show book not found. Therefore, it is strongly advised that the user type the book name exactly as it is shown in the library. After each run, the program will ask the user if he or she wants to keep searching for another book. The user simply needs to type “yes”, if he or she wants to continue searching. If the user wants to stop searching, he or she simply needs to type “no”. The program displays the entire library after the program terminates. The program also displays the library after a book is added to the library, in order to show that the book is indeed added to the library. Lastly, the program will display the book requested by the user, if the book is found.

Logical Exceptions

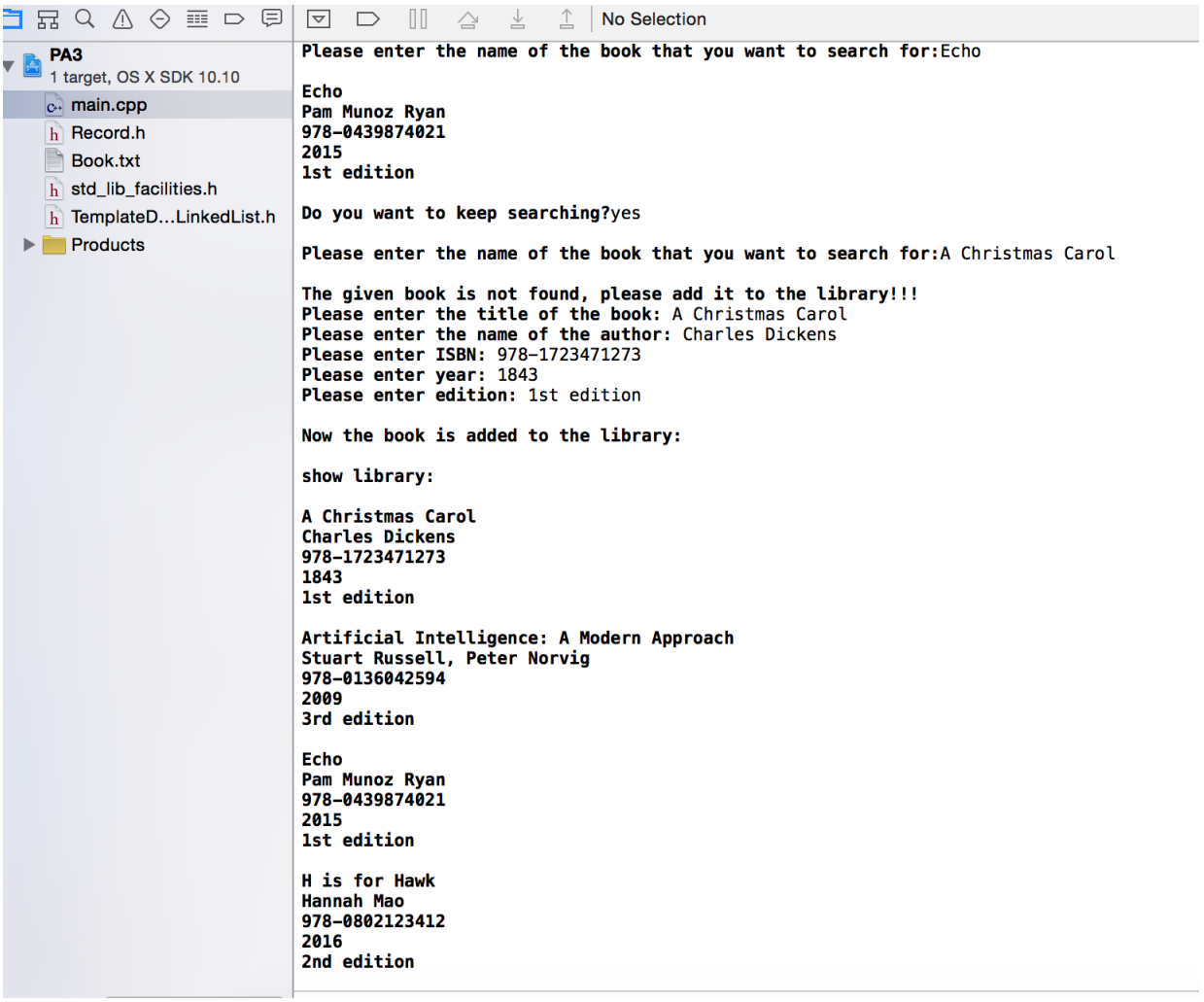
For Doubly Linked List class, there should be EmptyDLinkedList exceptions thrown for the following member functions: int first(), int last(), int removeFirst(), int removeLast(). when an Doubly Linked list is empty, it cannot return an object in the first node or return an object in the last node. It cannot remove the first object, and it cannot remove the last object. Therefore, exceptions must be thrown under these conditions.

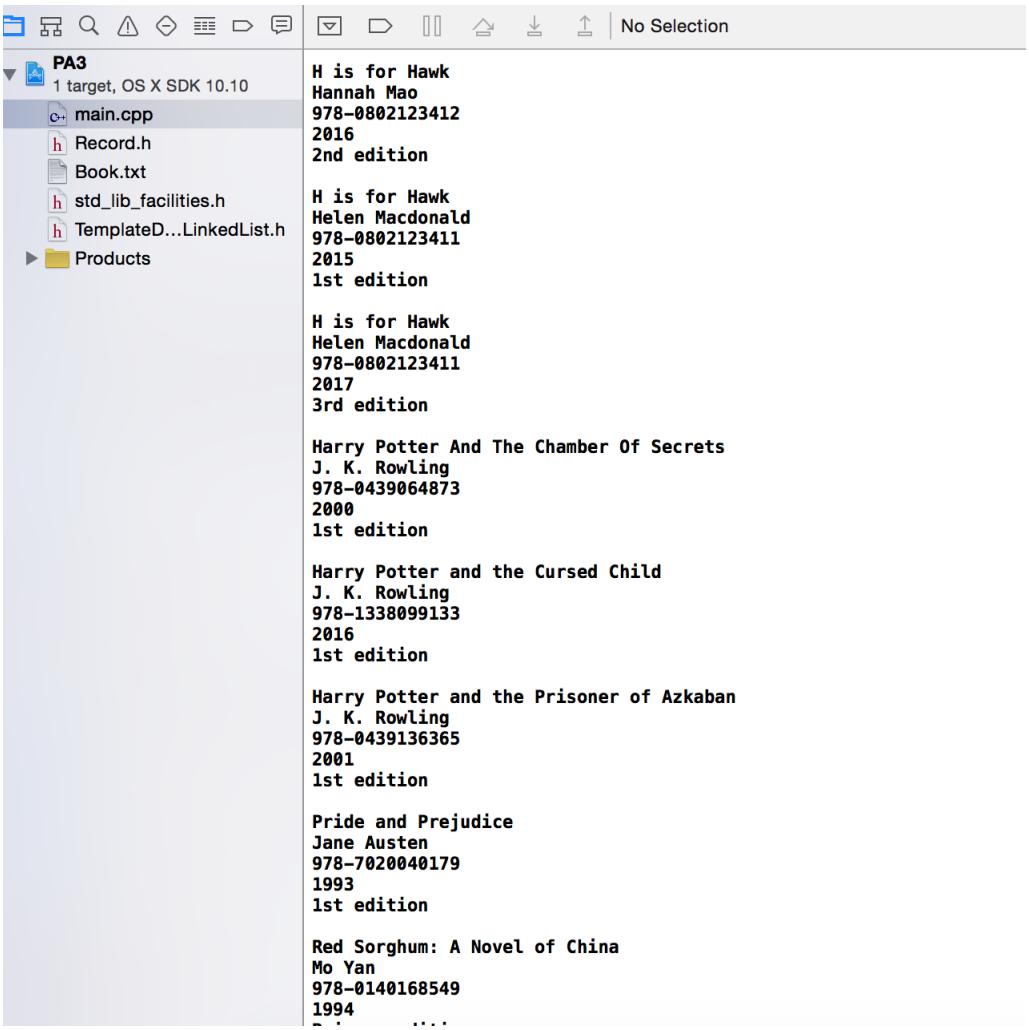
C++ object oriented or generic programming features

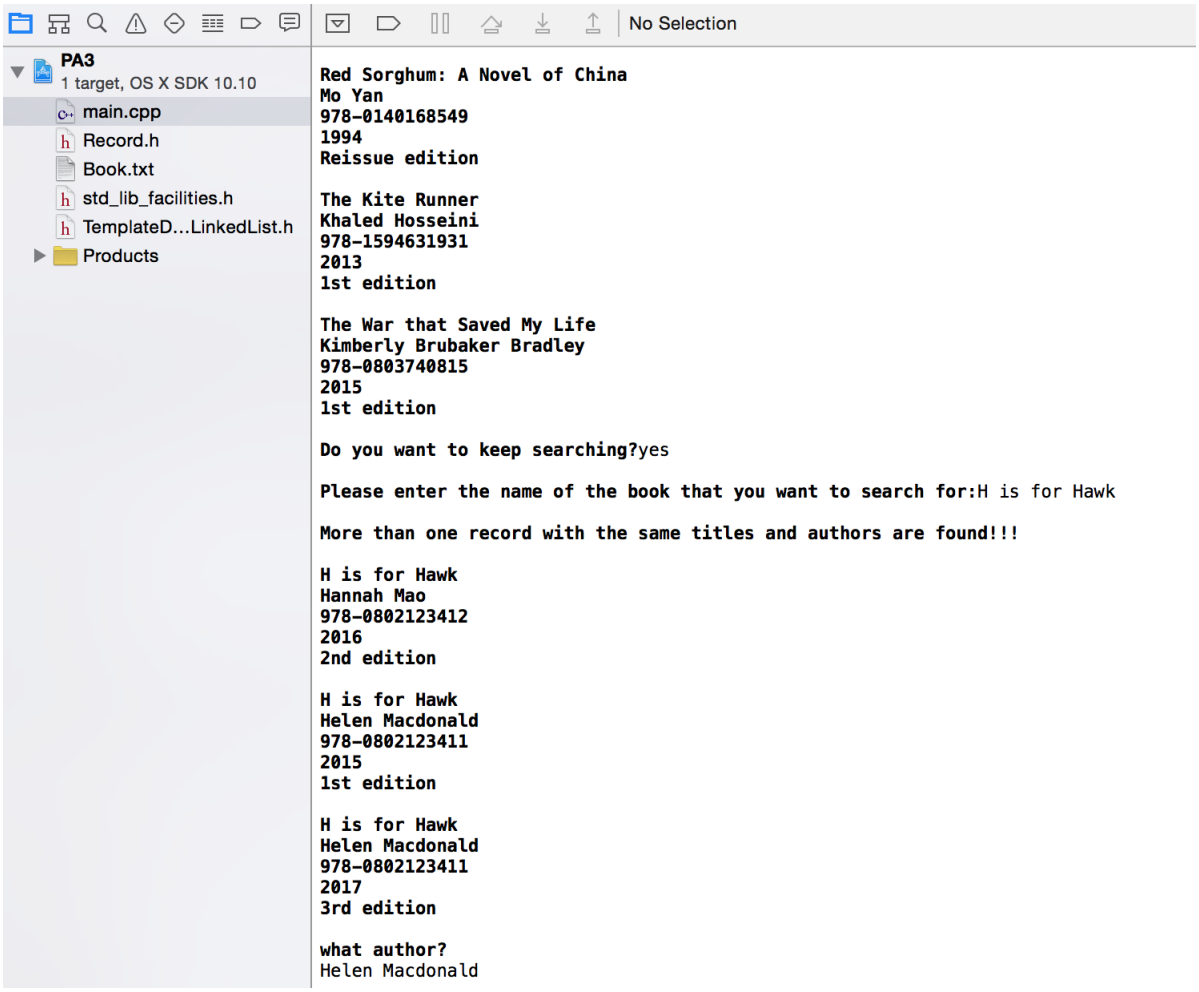
This assignment is object oriented because it requires writing implementations for classes. For part one, the students are required to write implementations for SimpleLinkedList class and DoublyLinkedList class. For part two, students are required to write a class Record which holds 5 items, and students are required to write overloading input operator, overloading output operator, and overloading operator less than for the Record class. The generic programming feature in this assignment is the templateDoublyLinkedList class, which is converted from DoublyLinkedList class. By using templateDoublyLinkedList class, we are able to have the doubly linked list to store elements with type Record.

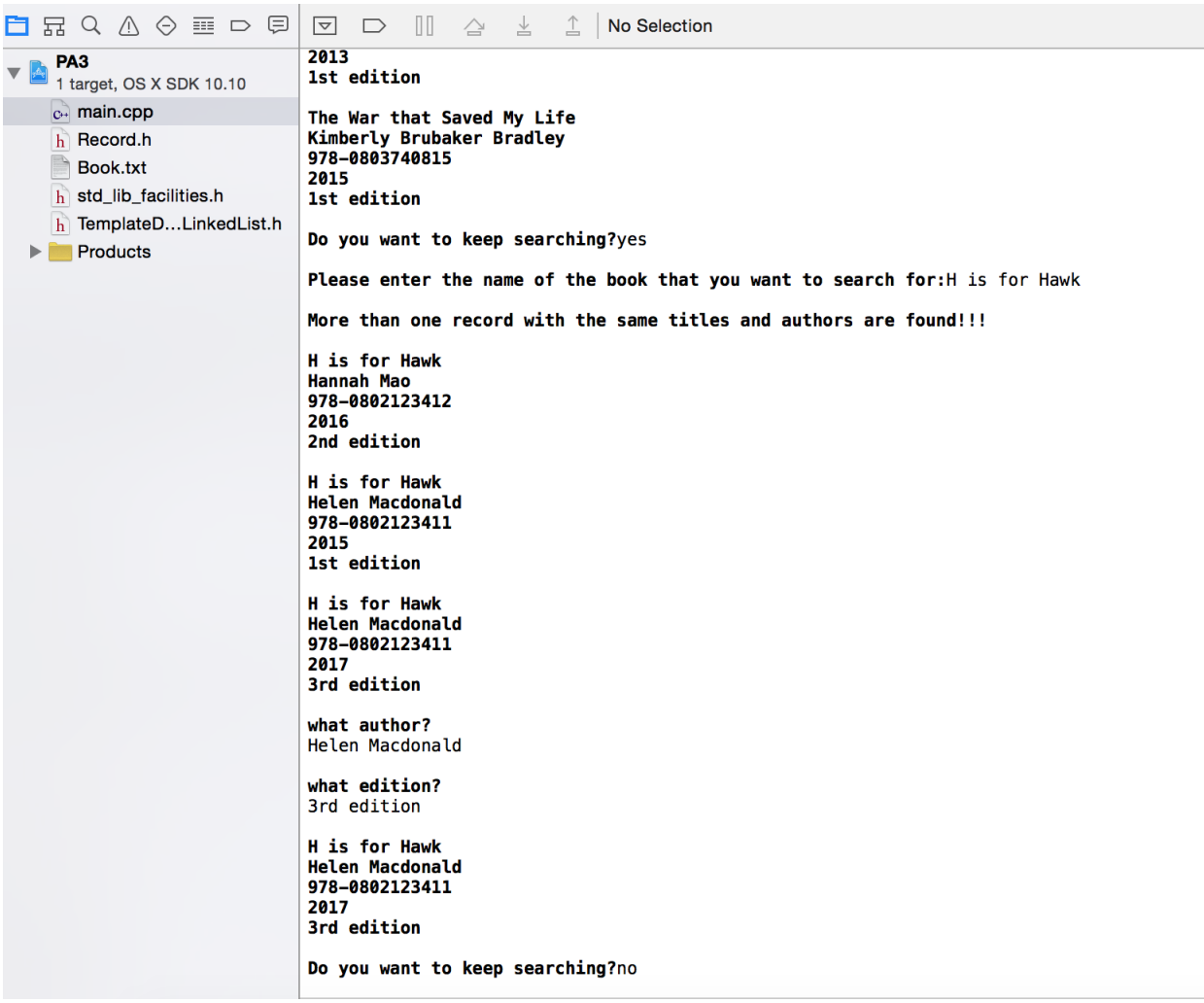
Tests

Part one of the testing is demonstrated in class by simply runnining the test files. Part two of the testing requires user input. There are three scenarios. The first scenario is that the user search for a book already existing in the library, the library will simply display the book that the user searches for. The second scenario occurs when the user searches for a book that is not already in the library, in this case, the user is required to input information about the book and the library will add the book to its database. The program will display the library showing the added book. The third scenario occurs when the user searches for a book, and more than one book has the same author and title. The library will first show the books that have the same titles, and the library will ask the user which author. After the user has chosen the author, the library will ask the user which edition of the book. After each search, the library prompts the user if he or she wants to keep searching. The user simply needs to type "yes" to continue searching, or "no" to terminate the program. The above three scenarios are tested below. Please see the results below.


No Selection
PA3
1 target, OS X SDK 10.10
main.cpp
Record.h
Book.txt
std_lib_facilities.h
TemplateD...LinkedList.h
Products
Please enter the name of the book that you want to search for:Echo
Echo
Pam Munoz Ryan
978-0439874021
2015
1st edition
Do you want to keep searching?yes
Please enter the name of the book that you want to search for:A Christmas Carol
The given book is not found, please add it to the library!!!
Please enter the title of the book: A Christmas Carol
Please enter the name of the author: Charles Dickens
Please enter ISBN: 978-1723471273
Please enter year: 1843
Please enter edition: 1st edition
Now the book is added to the library:
show library:
A Christmas Carol
Charles Dickens
978-1723471273
1843
1st edition
Artificial Intelligence: A Modern Approach
Stuart Russell, Peter Norvig
978-0136042594
2009
3rd edition
Echo
Pam Munoz Ryan
978-0439874021
2015
1st edition
H is for Hawk
Hannah Mao
978-0802123412
2016
2nd edition


No Selection
PA3
1 target, OS X SDK 10.10
main.cpp
Record.h
Book.txt
std_lib_facilities.h
TemplateD...LinkedList.h
Products
H is for Hawk
Hannah Mao
978-0802123412
2016
2nd edition
H is for Hawk
Helen Macdonald
978-0802123411
2015
1st edition
H is for Hawk
Helen Macdonald
978-0802123411
2017
3rd edition
Harry Potter And The Chamber Of Secrets
J. K. Rowling
978-0439064873
2000
1st edition
Harry Potter and the Cursed Child
J. K. Rowling
978-1338099133
2016
1st edition
Harry Potter and the Prisoner of Azkaban
J. K. Rowling
978-0439136365
2001
1st edition
Pride and Prejudice
Jane Austen
978-7020040179
1993
1st edition
Red Sorghum: A Novel of China
Mo Yan
978-0140168549
1994


No Selection
PA3
1 target, OS X SDK 10.10
main.cpp
Record.h
Book.txt
std_lib_facilities.h
TemplateD…LinkedList.h
Products
Red Sorghum: A Novel of China
Mo Yan
978-0140168549
1994
Reissue edition
The Kite Runner
Khaled Hosseini
978-1594631931
2013
1st edition
The War that Saved My Life
Kimberly Brubaker Bradley
978-0803740815
2015
1st edition
Do you want to keep searching?yes
Please enter the name of the book that you want to search for:H is for Hawk
More than one record with the same titles and authors are found!!!
H is for Hawk
Hannah Mao
978-0802123412
2016
2nd edition
H is for Hawk
Helen Macdonald
978-0802123411
2015
1st edition
H is for Hawk
Helen Macdonald
978-0802123411
2017
3rd edition
what author?
Helen Macdonald


No Selection
PA3
1 target, OS X SDK 10.10
main.cpp
Record.h
Book.txt
std_lib_facilities.h
TemplateD...LinkedList.h
Products
2013
1st edition
The War that Saved My Life
Kimberly Brubaker Bradley
978-0803740815
2015
1st edition
Do you want to keep searching?yes
Please enter the name of the book that you want to search for:H is for Hawk
More than one record with the same titles and authors are found!!!
H is for Hawk
Hannah Mao
978-0802123412
2016
2nd edition
H is for Hawk
Helen Macdonald
978-0802123411
2015
1st edition
H is for Hawk
Helen Macdonald
978-0802123411
2017
3rd edition
what author?
Helen Macdonald
what edition?
3rd edition
H is for Hawk
Helen Macdonald
978-0802123411
2017
3rd edition
Do you want to keep searching?no