CSCE 221 Cover Page
Homework Assignment

First Name:	Ryan	Last Name:	Parker	UIN:	327001603
User Name:		E-mail address:	ryanmparker@tamu.edu		

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more in the Aggie Honor System Office http://aggiehonor.tamu.edu/

Type of sources					
People					
Web pages (provide URL)					
Printed material					
Other Sources					

I certify that I have listed all the sources that I used to develop the solutions/code to the submitted work.

"*On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*"

Your Name (signature) Ryan Parker Date 6/8/2020

We were tasked with creating a library system. In order to create this system we were to create a class that stores information about a record and several implementations of linked lists.

A linked list is a data structure that consists of nodes that contain some sort of data and a pointer to the next node in the list. If it is a doubly-linked list then the node also contains a pointer to the previous node. Linked lists normally contain a head and a tail to know where the list begins and ends.

The Record class was implemented with setter and getter methods for title, author, ISBN, publishing year, and edition. Then the <<, >>, and == operators were overloaded to function with Records. The << operator is implemented by outputting the different member variables related to book information. The >> operator is implemented by pulling 5 lines from input and assigning each line to a different member variable. The == operator checks if the titles are the same and if they are checks if the author is the same and then checks if the ISBN is the same. If any of them do not match then the operator evaluates to false.

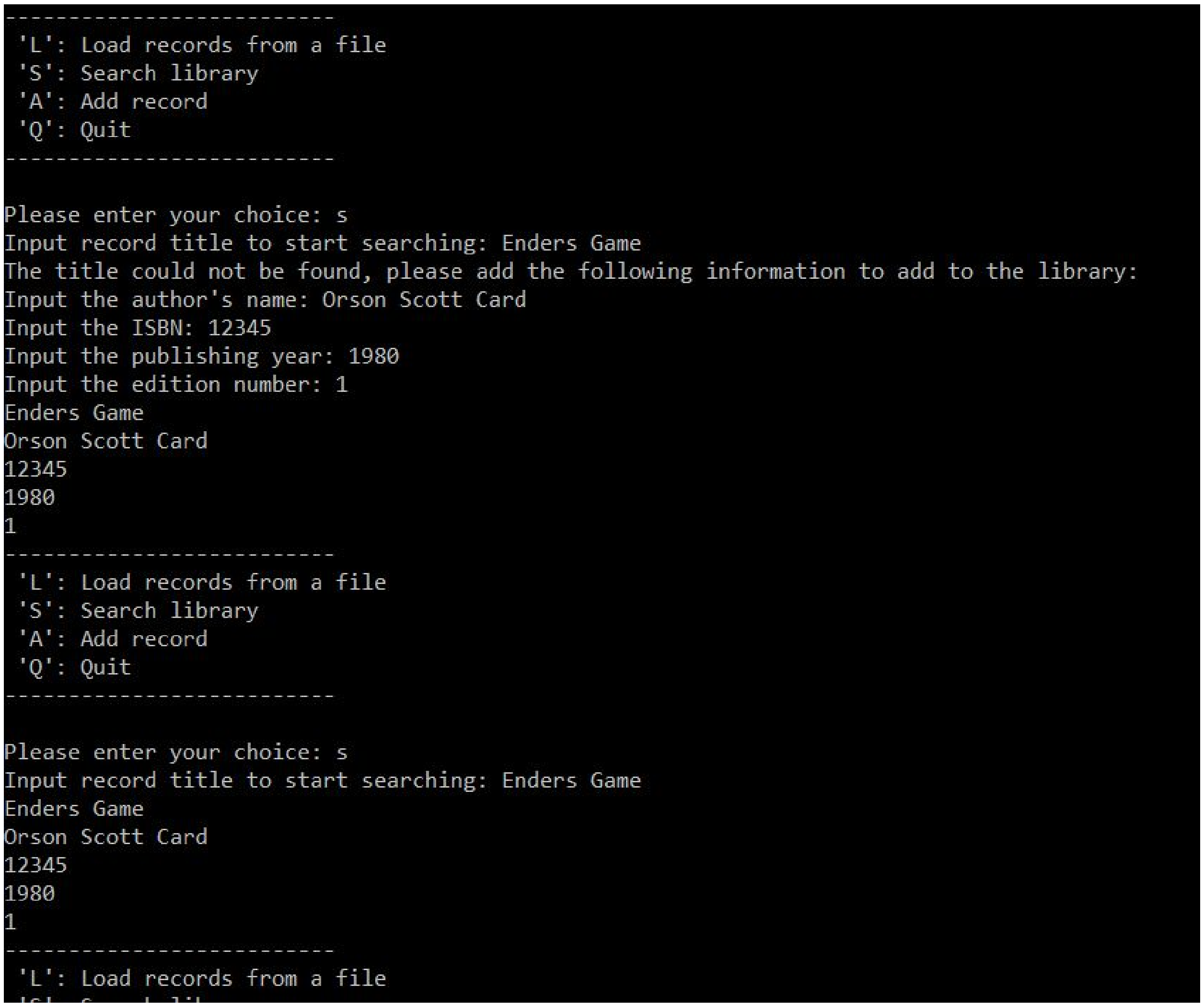
The linked lists are implemented with a Node struct to hold the data and pointers for each node. The linked list class has functions for adding and removing nodes to the beginning, end, and before or after another node. It also has functions that will check if the list is empty and to return the first and last nodes and the data those nodes contain. The functions involving adding or removing nodes all functioned in O(1) time. The destructor and move and copy assignment operators functioned in O(n) time as they had to loop through to delete the list.

To build the library itself I implemented a function to search the database, one to sort it, one to know which doubly-linked list to search through, and a way to add new records to the library. To sort I used a bubble sort algorithm that runs in O(n^2). I initially implemented selection sort and then insertion sort before settling on bubble sort due to trying to solve an issue of improper sorting. It turns out there an issue in another part of the program causing the sorting to be wrong but I saw no need to reimplement a different sorting algorithm. To search the library I used a linear search function that runs in O(n). Creating new records and knowing which DLL to search both run in O(1). I also implemented a function to load a library from a file to be used for testing purposes that runs in O(n).

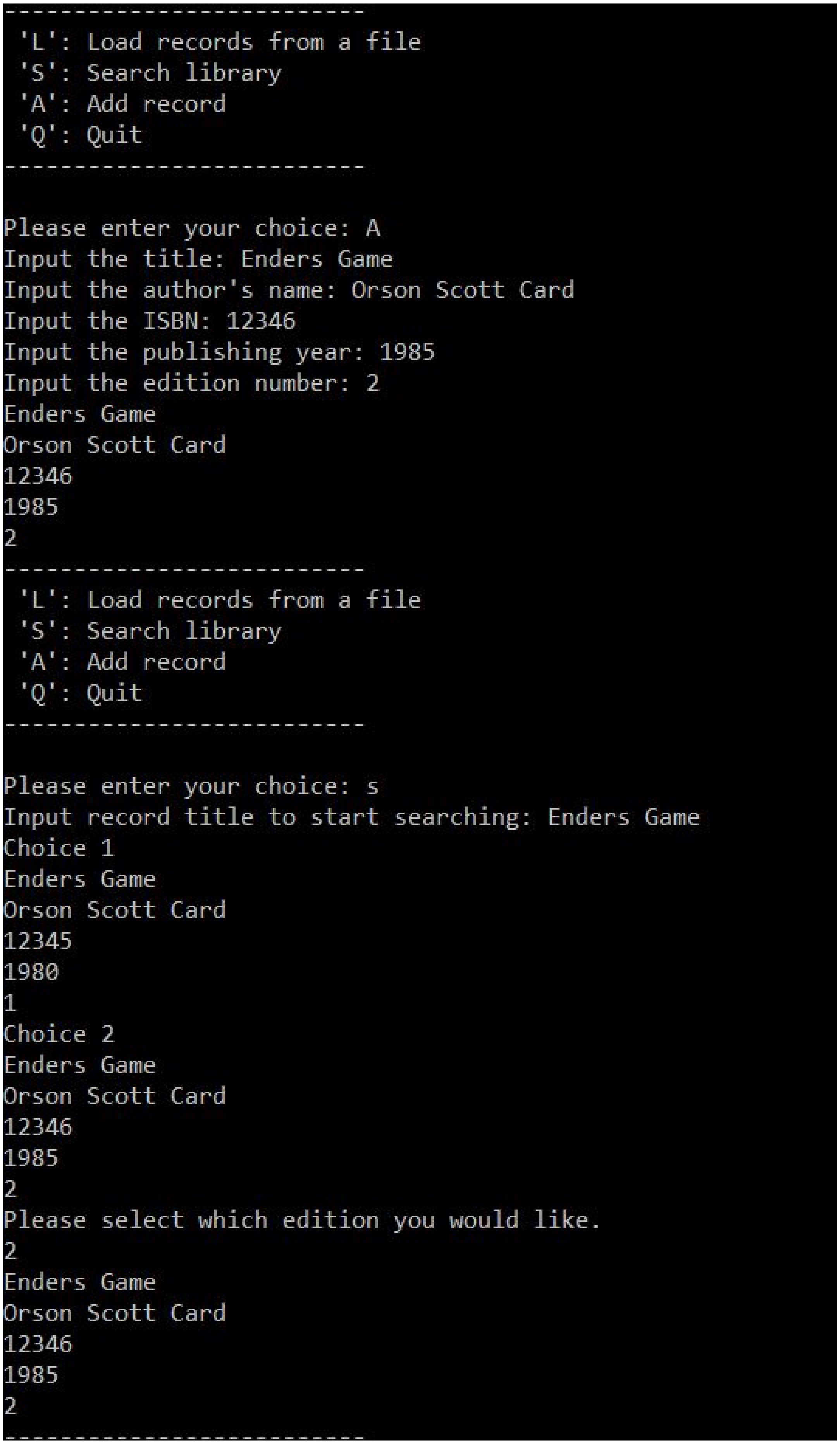
For a library of size n the running time function to search would be f(n)=3n+2 and O(n) and inserting a new record to a library of size n would be f(n)=6n^2+9 and O(n^2) because when adding a new record I sort the doubly-linked list it is in.

The classes are called Record, DLList, SimpleDLList, TemplatedDLList. The program used a templated implementation of a doubly-linked list to create a doubly-linked list of records.

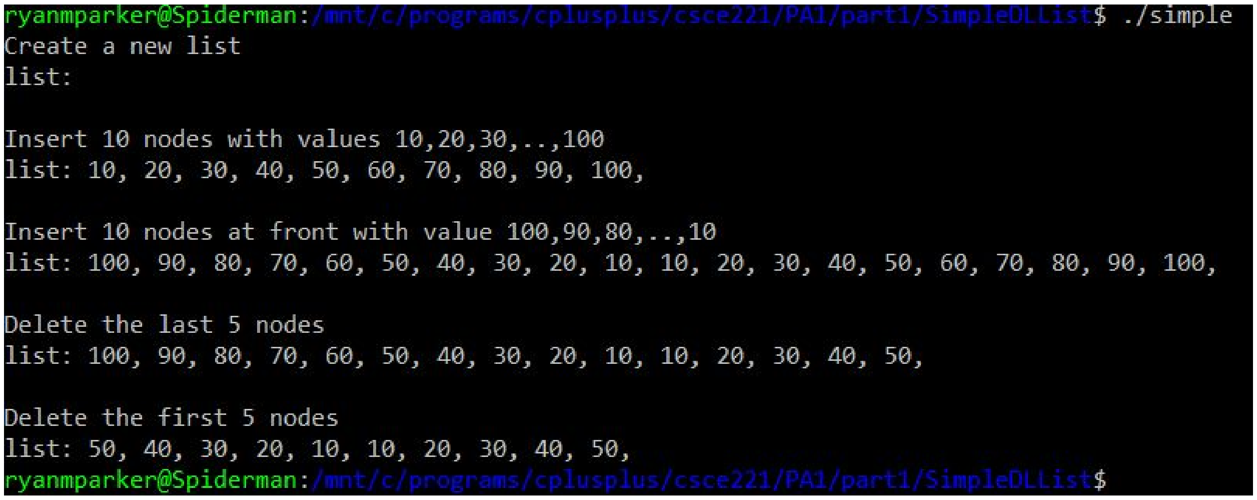
Each part is in its own folder labeled accordingly. For part 0 the file you need to run is called pa1 and the class is implemented in Record.h. For simple doubly-linked list in part 1 you need to run simple. DLList you need to run run-dll and the templated version run run-tdll. For part 2 the new functions are implemented in main.cpp and you need to run the file called library to run the program. It takes input from the keyboard for part 2. Part 0 takes input from Books.txt.



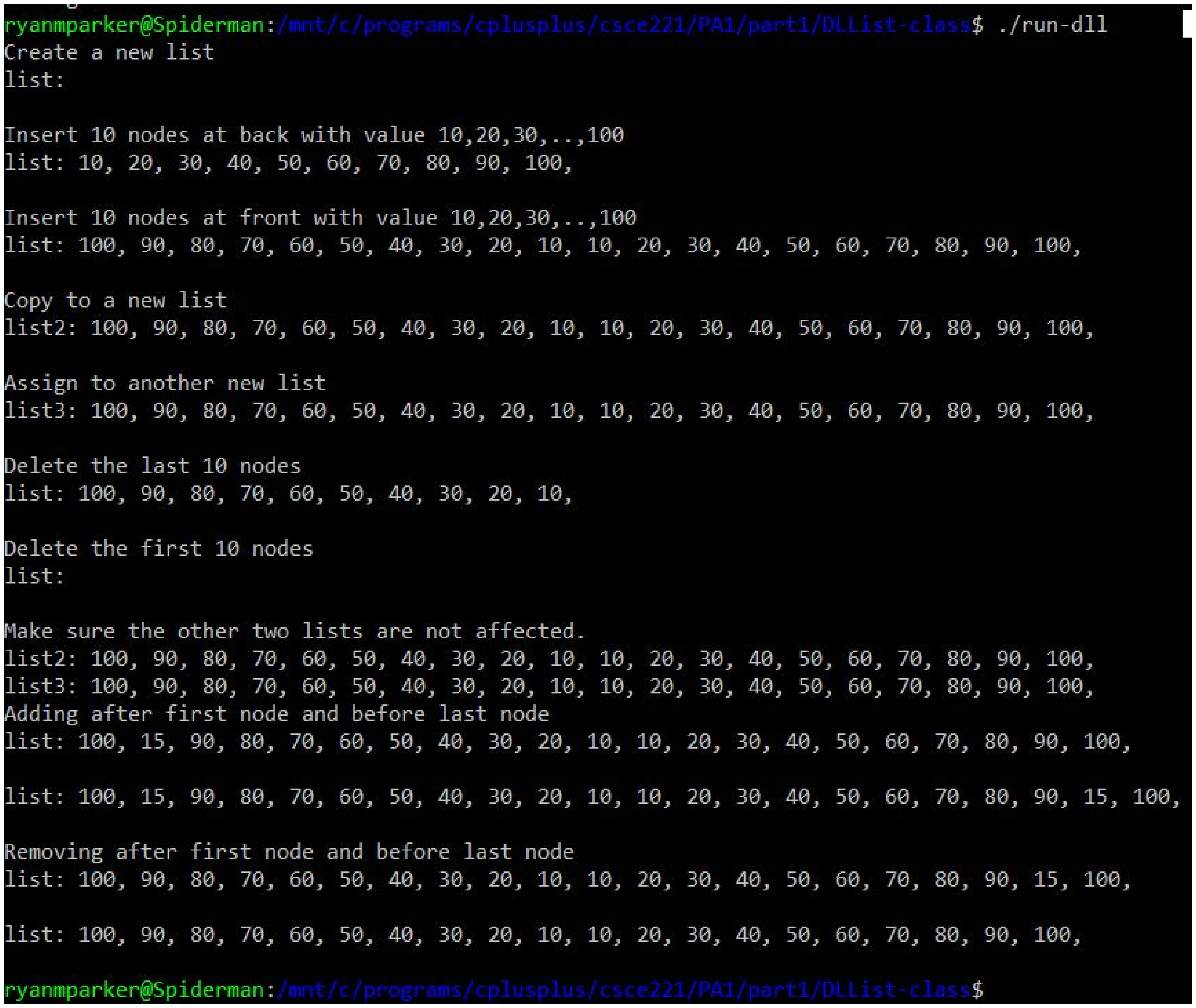
```
--------------------------
 'L': Load records from a file
 'S': Search library
 'A': Add record
 'Q': Quit
--------------------------

Please enter your choice: s
Input record title to start searching: Enders Game
The title could not be found, please add the following information to add to the library:
Input the author's name: Orson Scott Card
Input the ISBN: 12345
Input the publishing year: 1980
Input the edition number: 1
Enders Game
Orson Scott Card
12345
1980
1
--------------------------
 'L': Load records from a file
 'S': Search library
 'A': Add record
 'Q': Quit
--------------------------

Please enter your choice: s
Input record title to start searching: Enders Game
Enders Game
Orson Scott Card
12345
1980
1
--------------------------
 'L': Load records from a file
```



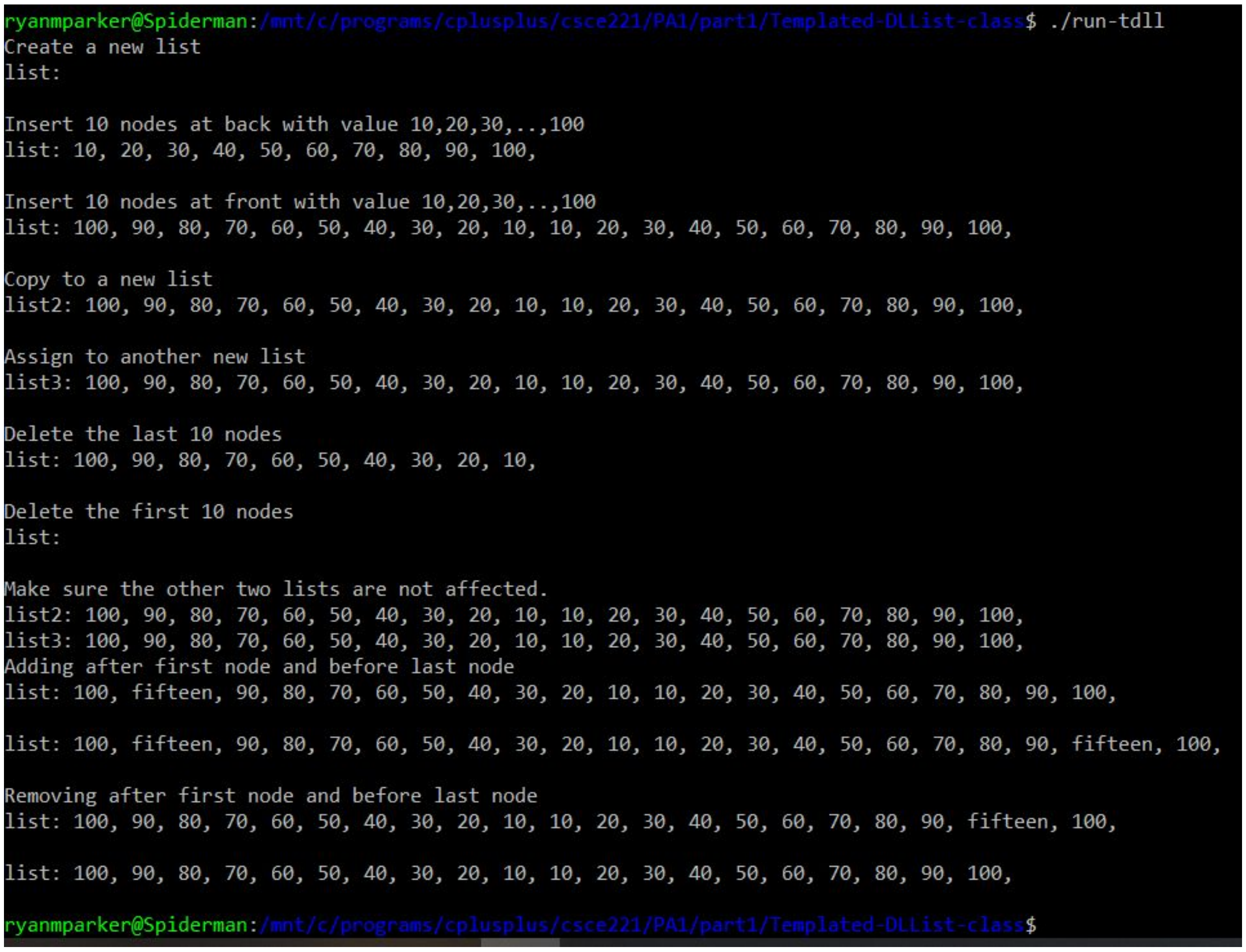
```
--------------------------
 'L': Load records from a file
 'S': Search library
 'A': Add record
 'Q': Quit
--------------------------

Please enter your choice: A
Input the title: Enders Game
Input the author's name: Orson Scott Card
Input the ISBN: 12346
Input the publishing year: 1985
Input the edition number: 2
Enders Game
Orson Scott Card
12346
1985
2
--------------------------
 'L': Load records from a file
 'S': Search library
 'A': Add record
 'Q': Quit
--------------------------

Please enter your choice: s
Input record title to start searching: Enders Game
Choice 1
Enders Game
Orson Scott Card
12345
1980
1
Choice 2
Enders Game
Orson Scott Card
12346
1985
2
Please select which edition you would like.
2
Enders Game
Orson Scott Card
12346
1985
2
--------------------------
```



```
ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/SimpleDLList$ ./simple
Create a new list
list:

Insert 10 nodes with values 10,20,30,..,100
list: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Insert 10 nodes at front with value 100,90,80,..,10
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Delete the last 5 nodes
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50,

Delete the first 5 nodes
list: 50, 40, 30, 20, 10, 10, 20, 30, 40, 50,
ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/SimpleDLList$
```



```
ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/DLList-class$ ./run-dll
Create a new list
list:

Insert 10 nodes at back with value 10,20,30,..,100
list: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Insert 10 nodes at front with value 10,20,30,..,100
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Copy to a new list
list2: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Assign to another new list
list3: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Delete the last 10 nodes
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10,

Delete the first 10 nodes
list:

Make sure the other two lists are not affected.
list2: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
list3: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
Adding after first node and before last node
list: 100, 15, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

list: 100, 15, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 15, 100,

Removing after first node and before last node
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 15, 100,

list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/DLList-class$
```



```
ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/Templated-DLList-class$ ./run-tdll
Create a new list
list:

Insert 10 nodes at back with value 10,20,30,..,100
list: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Insert 10 nodes at front with value 10,20,30,..,100
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Copy to a new list
list2: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Assign to another new list
list3: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

Delete the last 10 nodes
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10,

Delete the first 10 nodes
list:

Make sure the other two lists are not affected.
list2: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
list3: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,
Adding after first node and before last node
list: 100, fifteen, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

list: 100, fifteen, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, fifteen, 100,

Removing after first node and before last node
list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, fifteen, 100,

list: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100,

ryanmparker@Spiderman:/mnt/c/programs/cplusplus/csce221/PA1/part1/Templated-DLList-class$
```