CSCE 221 Cover Page

Programming Assignment#3

Due by March 7 midnight to eCampus

First name: Aldo Last Name: Leon Marquez UIN: 326004699

User Name: allema98 E-mail address: allema98@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: Aggie Honor System Office

Types of Sources			
People	TA sec 501		
Web Pages	https://piazza.com	Stack Overflow	
Printed material	-		
Other Sources	Lecture Slides		

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.
"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Name : Aldo Leon Marquez Date: 03/07/2019

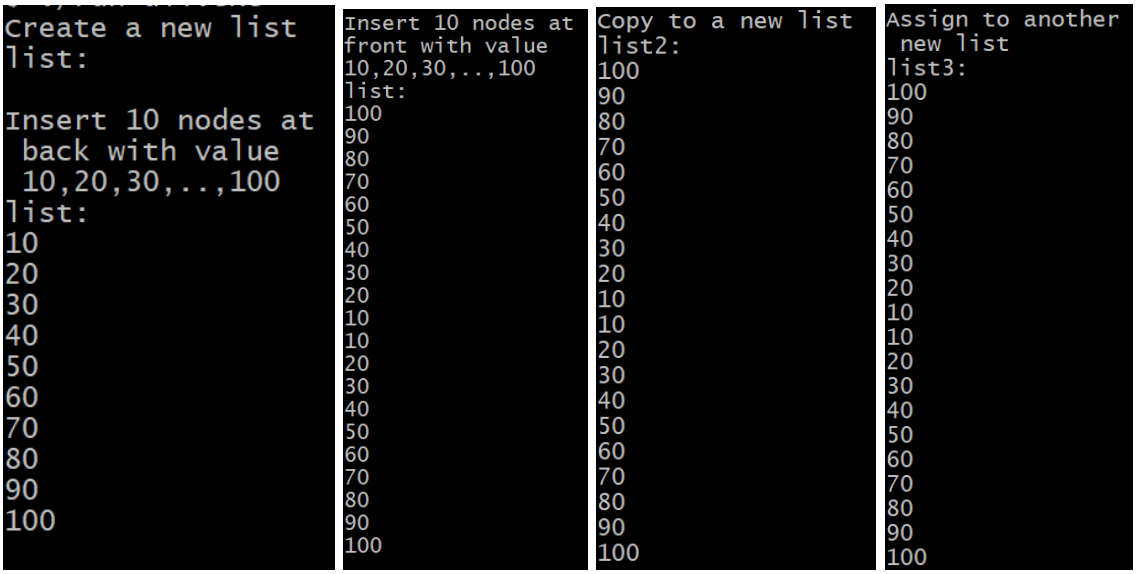
Part1

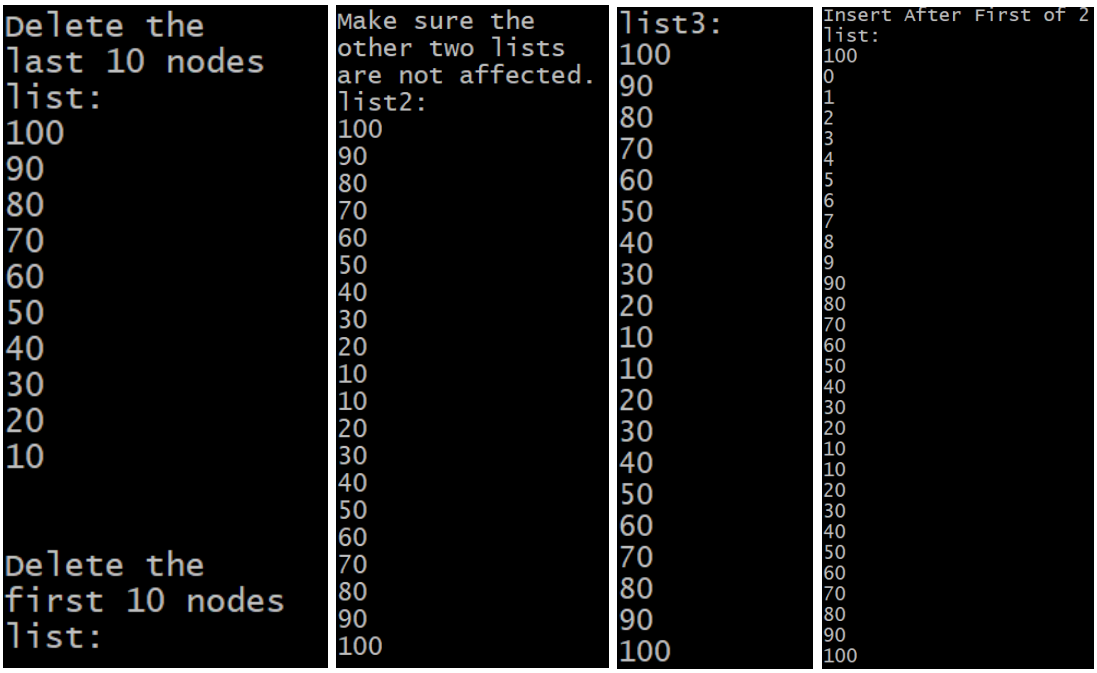
Doubly Linked List Implementation

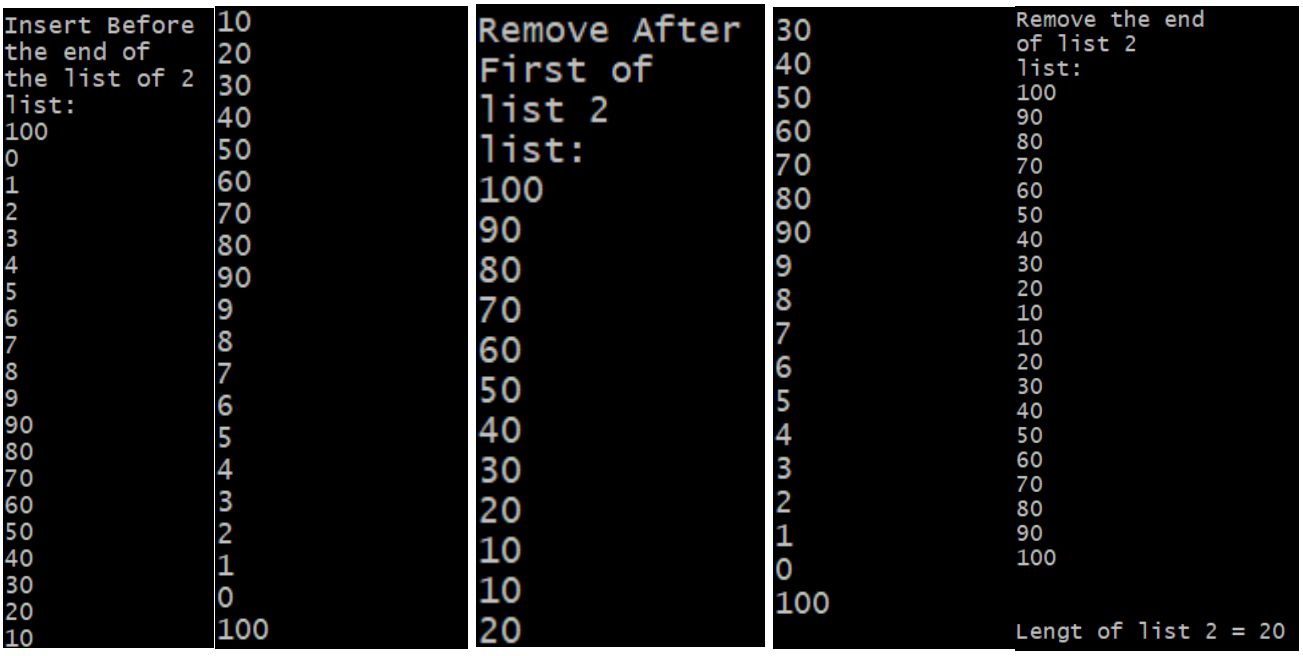
For this assignment I used the concept of a linked listed. A linked list is a form of storing a list of values (or nodes with more than one value) using pointers. The Way a Doubly Linked List Works is by first creating a Two refence nodes that serve as the beginning and the ending of the list. These nodes also contain a pointer to the next node or previous node(respectively). Nodes will later be added or removed by simply connecting and the pointers to the new node, making sure both previous and next pointers are correctly assigned, or a node will be removed by taking care of the pointers to said node and then deallocating the memory used by the node.

For this Assignment a generic Implementation of a Doubly Linked List is coded, the implementation contains all basic function required for a doubly linked list to work. It will be teste using a “main.cpp” file for the regular implementation, then a “Templatemain.cpp” for the generic Implementation.

Doubly Linked list (Regular and Templated) Test Cases proof







Note:Screenshots added for the non templated test cases, since bith test cases shared the same input values , but for the templated one used strings as varieble type. Test case file included in Phase folde(Templatedmain.cpp)

The time Complexoity for the class function are as follows, with the longest one being whenever the whole list might be traversed

Function	Complxity in terms of Big O
Constructor	O(1)
Copy Constructor	O(n)
Move Constructor	O(1)
Destructor	O(1)
Assignment Operator	O(n)
Move Assignment	O(1)
Get First Pointer	O(1)
Get After Last Pointer	O(1)
Is Empty ?	O(1)
First value	O(1)
Last Value	O(1)
Insert First	O(1)
Insert Last	O(1)
Remove First	O(1)
Remove Last	O(1)
Insert After desired node	O(1)
Insert Before desired node	O(1)
Remove After desired node	O(1)
Remove After desired node	O(1)
Length of the List	O(n)

Part 2

Min Queue Implementation

For the second pat of the Assignment a new secondar class will be implemented. The concept of a MinQueue consist in queueing up a set mount of values into a temporary list, that when emptied of whne poping a single value the min value of the list Queue will be returned.

Test Cases for MinQueue

```
Create a new list
list:

insert 10 random
numbers to a list
Full Queue
Size of Queue: 10
7
3
7
1
9
10
3
4
4
10
Minimum from the Queue: 1
Empty Queue
Size of Queue: 0
```

```
Char Queue
Full Char Queue
f
q
l
q
c
q
r
t
o
w
Size of Queue: 0
Minimum from the Queue: c
Empty Queue
Size of Queue: 0
Dequeue of an
empty Queue
Empty Queue
```

We have a similr case with the time complexity of the Minqueue Class, were only a few functions have to go through all of the list to get the min value, or to count the total sixe of the Queue

Function	Time Complexity in terms of Big O
Constructors	O(1)
Enqueue (unsorted)	O(1)
Dequeue(min value is dequeue)	O(n)
Size	O(n)
Min(return only not pop)	O(n)