Jared Westmoreland
Section# 002
Lab# 302

The real difficult is in my function.cpp and classfile.h, everything else was only routine for the user. The requirements for this homework were to make a linked list with the functions add, remove, search, print, command, exit.

I learned how to create a node and it was through struct and putting the same struct name within the struct. I put a character value in it and sealed it. Next it was classes, I decided to make it easy just to create everything public; I do know it's not a good thing to make your variables within public because others might use you variable by accident, but because I'm only working on it I think it's fine if I don't use private. I also put all of my functions within it, empty, addFirstElement, add, search, remove, and print, command.

Starting with add I referred to the book and had difficulty putting words into code and there was an example of add, but it was not quite what I needed. To create the first head I needed a check to see if there was any information within the linked list. So I made a made empty to check whether or not there was a head existing. If not it return true, if so it return false. In add, if it were true it would go to addFirstElement and create the first head. After that if the user inputted another character/number it would go through a couple of hoops.

Next was search. It receives the user input and then checks if there is a head. Afterwards, it goes through a while loop and compares it to all the nodes and checks if it matches. If it does it returns true and if it does not it returns false.

Next is remove. It receives the user input and then checks if there is a head. Afterwards, it goes through a loop comparing it to the all the node with the user input. If it matches it deletes it.

Finally, we have print. Checks if there is a head, if not it props "list is empty". Else it displays all he nodes with a while loop.

Add

Get user input

Check if head
Yes- continue
No- go to different function, Add 2

If yes
Compare the value of each

If it's greater
Put it above
If it's less
Put it below
If it's the same value
Cout<<"already exists\n";

Search

Get user input

Check if head
If no
List is empty
If yes
Go through loop
Check each node for user input
If it does cout<<"true\n";
If it doesn't cout<<"false\n";

Remove

Get user input

Check if head
If no
List is already empty
Go through loop
Look at every value and compare it to user input
If it does delete it
If it does not cout<<"Input does not exist\n";

Print

Check if head
If no
List is empty
If yes
Loop and cout evey node

Command

Cout the menu plus the command prompt