Name:
Advanced Programming in C++
Lab Exercise 4/6/2022

Introduction to Linked Lists

In this exercise, we will be working with a NumberList class. You will find a copy of this class (NumberList.h and NumberList.cpp) in the project:

1. Write a program that will create a list of 10 nodes containing 5.6, 7.5, 12.8, 34.8, 24.2, 78.1, 78.5, 112.9, 31.6, and 44.5 in ascending order. You must add the values to the list in the above order. Call the displayList function to display your list. You should write a main function to exercise the NumberList class.

2. Remove the nodes containing 112.9 and 24.2 from your list and re-display the list.

3. Modify the NumberList class to add a prependNode function which inserts a node at the beginning of the list. Write a main function to test your new class.

4. Write displayListBackwards function that will display your linked list backwards. Write a main function that will test this new method.

Here is the code for prependNode.

```
void NumberList::prependNode(double num)
{
        ListNode *newNode;

        // Allocate a new node & store num
        newNode = new ListNode;
        newNode->value = num;
        newNode->next = NULL;

        // If there are no nodes in the list
        // make newNode the first node
        if (!head)
                head = newNode;
        else    // Otherwise, insert newNode at end
        {
                //point the new node to the location of the first node in list
                newNode -> next = head;

                //make the head pointer point at the new node
                head = newNode;
        }
}
```

Here is the code for displayListBackwards()

```
void NumberList::displayListBackwards()
{
        ListNode *nodePtr;
        vector<double> temp;  //create a vector to hold values
        int i;

        //set nodePtr to beginning of the list
        nodePtr = head;

        //Copy each element in the list to the vector
        while (nodePtr != NULL)
        {
                temp.push_back(nodePtr -> value);
                nodePtr = nodePtr -> next;
        }

        //Display vector of list values from end to start
        for (i = temp.size() - 1; i > 0; i--)
                cout << temp[i] << " <-- ";

        //display the first element without the <-- symbol
        cout << temp[i] << endl;
}
```

Print a sample output of your final program and send it to me.