Name
Advanced Programming in C++
Lab Exercise 4/5/2024

In this lab you will learn about the Stack and Queue classes. A stack is a data structure that is known as LIFO (Last In First Out). A queue is a data structure that is FIFO (First In First Out). These data structures are used extensively in Computer Science. One application of these data structures is within the operating system software. Queue data structures are used as process schedulers for example. Stacks are also used by operating systems to store process state when a process is preempted by the operating system scheduler or an interrupt. Do not worry if you do not understand these concepts now. These topics are normally covered in the third year of a Computer Science curriculum.

Your task today is to write and test a Stack and Queue class. Your Stack and Queue will use the NumberList class. I have provided the code for the Stack and the Queue classes definitions (mystack.h, myqueue.h). I have also included some driver programs to test your classes. The Stack class adds and removes data from the stack by calling the push and pop functions. The Queue class adds and removes data from the queue by using the functions enqueue and dequeue. The implementation that I chose for my Stack and Queue classes is the Linked List. In a stack you add and remove from the same end of the list. You will need to modify your NumberList class which so that it has a prependNode function to add a node at the beginning of the list (push)and a getNode function to remove a node from the beginning of the list (pop). To implement a Queue you add and remove from different ends of the list. You can use the appendNode to add a node to the end of the list (enqueue) and getNode to remove a node from the beginning of the list (dequeue). It should be noted that before you pop a stack or dequeue a node from a queue, you should check if the stack or queue is empty.

Create two separate projects and get my Stack Demo and Queue Demo programs working. You will need to add you NumberList class to your project. Turn in a copy of your source code as well as a capture of your output attached to this sheet.

```
//Test program to test the MyStack class
#include <iostream>
using namespace std;
#include "mystack.h"

int main()
{
	MyStack test;
	test.push(12.7);
	test.push(15.7);
	test.push(22.7);
	test.push(32.7);
	test.display();
	cout << test.pop() << " popped" << endl;
	test.display();
	cout << test.pop() << " popped" << endl;
	test.display();
	cout << test.pop() << " popped" << endl;
	test.display();
	cout << test.pop() << " popped" << endl;
	test.display();
	cout << test.pop() << " popped" << endl;
	test.display();
	return 0;
}

//MyStack class definition (mystack.h)
#ifndef MYSTACK_H
#define MYSTACK_H
#include "NumberList.h"

class MyStack
{
	private:
		NumberList stack;
		int size;
	public:
		MyStack();
		void push(double);
		double pop();
		bool checkEmpty();
		void display();
};
#endif
```

```
//Test program to test the MyQueue class
#include <iostream>
using namespace std;
#include "myqueue.h"

int main()
{
	MyQueue test;
	test.enqueue(12.7);
	test.enqueue(15.7);
	test.enqueue(22.7);
	test.enqueue(32.7);
	test.display();
	cout << test.dequeue() << " dequeued" << endl;
	test.display();
	cout << test.dequeue() << " dequeued" << endl;
	test.display();
	cout << test.dequeue() << " dequeued" << endl;
	test.display();
	cout << test.dequeue() << " dequeued" << endl;
	test.display();
	cout << test.dequeue() << " dequeued" << endl;
	test.display();
	return 0;
}

//MyQueue class definition (myqueue.h)
#ifndef MYQUEUE_H
#define MYQUEUE_H
#include "NumberList.h"

class MyQueue
{
	private:
		NumberList queue;
		int size;
	public:
		MyQueue();
		void enqueue(double);
		double dequeue();
		bool checkEmpty();
		void display();
};
#endif
```