Data Structures and Fundamentals of Programming

Problem #1

In C++ implement a generic class, called `Queue<T>`, that uses a single-linked list implementation. It implements the queue abstract data type (ADT). It must be generic on the type of the data to be stored. Give all class declaration(s) and implement the following for `Queue`:

- Default constructor
- Destructor
- Copy-constructor
- Swap that runs in constant time no matter what the length of the queues
- Assignment operator – using standard copy semantics
- `enqueue(T)` – takes a parameter of type T and adds it to the queue
- `T dequeue()` – removes an item from the queue

Note: Your implementation can NOT use STL or any other libraries (standard or otherwise).

Problem #2

In C++ implement a generic double-linked-list class, called List<T>, that uses dynamic memory allocation. The list must behave as following:

frontptr -> X_1 <->X_2 <-> … <-> X_n <- backptr

where X_1 is the first node in the list and X_n is the last node in the list. Besides List, you will need a class called node<T>. Along with the class declaration(s) you will need to implement a following member functions for List<T>:

- Default constructor
- Copy constructor
- Destructor
- `addToBack()` – Adds an item to the back of the list.
- `addToFront()` – Adds an item to the front of the list.
- `T Remove(node<T>*)` – removes a node from the list, given a pointer to the node.

Note: Your implementation can NOT use STL or any other libraries (standard or otherwise).

Problem #3

In C++ implement a ternary tree abstract data type (ADT) that uses dynamic memory allocation. Make it a tree of integers. Each node will have between 0 and 3 children (left, middle, right). Along with the class declaration(s), you must implement the following methods for the class ternary:

- Default constructor
- Destructor – must be recursive or use a recursive method to delete all the nodes in a tree.
- Copy-constructor – must be recursive or use a recursive method make a complete copy of a tree.
- Preorder – which prints out the entire tree using a preorder traversal. Must be recursive.
- Postorder – which prints out the entire tree using a postorder traversal. Must be recursive.

Note: Your implementation can NOT use STL or any other libraries (standard or otherwise).