

Data Structures and Fundamentals of Programming

Problem #1

In C++ implement a **generic** class, called `Queue<T>`, that uses a **single-linked list** implementation. This should implement the queue ADT. It should be generic on the type of the data to be stored. Give all class definitions and implement the following for `Queue`:

- Default constructor
- Destructor
- Copy-constructor
- Assignment operator
- `enqueue (T)` – takes an parameter of type T and puts it on the end of the queue
- `T dequeue ()` – removes a node from front of the queue

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

Problem #2

When constructing the ADT Binary Search Tree using a dynamic data structure a destructor, copy-constructor, and assignment operator would be needed. For this problem answer the following questions for these three methods:

- What traversal algorithms are most appropriate to implement each of these three methods?
- Give a pseudo-code (C++ like) description for each of these three methods and a short description of why the particular traversal algorithm should be used. You do NOT need to give a full implementation of the binary tree class.

Problem #3

Implement a function, in C++ to convert a **fully** parenthesized infix expression into postfix. You can assume the expression is correct and that only single character operands A-Z will be used. The infix expression will be passed into the function as a null terminating character array or string. The binary operators +, -, *, / with standard precedence are to be supported. You do not need to support unary operators. Additionally, you can assume that a generic class `Stack<T>` exists with `push` and `pop` defined as normal. You may also use the C++ string class to solve this problem.

```
string expr1 = "(A*((B+C)-D))";  
string expr2 = "(A+(B*(C+D)))";
```