

Data Structures and Fundamentals of Programming (Questions 5-8)

Problem 5

In C++ implement a **generic** singly-linked-list class, called `Stack<item>`, that uses dynamic memory allocation (supporting any size). `item` is the type of data stored in the stack. This should implement the stack ADT. The stack should look something like the following:

$$\text{TOS} \rightarrow X_1 \rightarrow X_2 \rightarrow \dots \rightarrow X_n$$

where X_1 is the node on the top of the stack and X_n is at the bottom of the stack. Besides `Stack`, you will most likely want another class or struct called `node`. Along with the class definition(s), you must implement the following methods for `Stack`:

- `Stack()` - Default constructor
- `~Stack()` - Destructor
- `Stack(const List(item&))` - Copy-constructor
- `push(item)` – takes an parameter of type `item` and creates a new node that is added to the front of the list.
- `item pop()` – removes a node from the end of a list and returns its contents.

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

Problem 6

For a given file, `F`, is the Huffman tree unique? If your answer is not, specify a file and show at least two Huffman trees (excluding a minor image) that give the optimal prefix code.

Problem 7

In the implementation of an ADT binary search-tree using a dynamic data structure you would need to build a a) destructor, b) copy-constructor, and c) assignment operator.

- What traversal algorithms are most appropriate to implement each of these three methods?
- Give a pseudo-code (C++ like) description of each of these three methods and a short description of why the particular traversal algorithm should be used.

Problem 8

Implement a function (using a stack based algorithm), in C++, to check if a given infix expression is correctly parenthesized. The expression uses C++ syntax. The expression can be stored as a simple null terminating char-string such as below or as a C++ string. Additionally, you can assume that a generic class `Stack` is defined as normal and you may also use the C++ string class.

```
char expr1[] = "(a+b)";
string expr22 = "((((a+b) * (c+d)) + 1))";
```