

# Data Structures and Fundamentals of Programming

## Problem #1

In C++ implement a **generic** `stack` class, using a linked list and dynamic memory allocation. It should be generic on the type of data stored in the `stack`. The `stack` data structure should look something like the following:

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

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

- Default constructor
- Destructor
- Copy-constructor
- `push` which takes a parameter of type `item` and creates a new node that is added to the top of the stack.
- `pop` which removes a node from the stack and returns its contents.

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

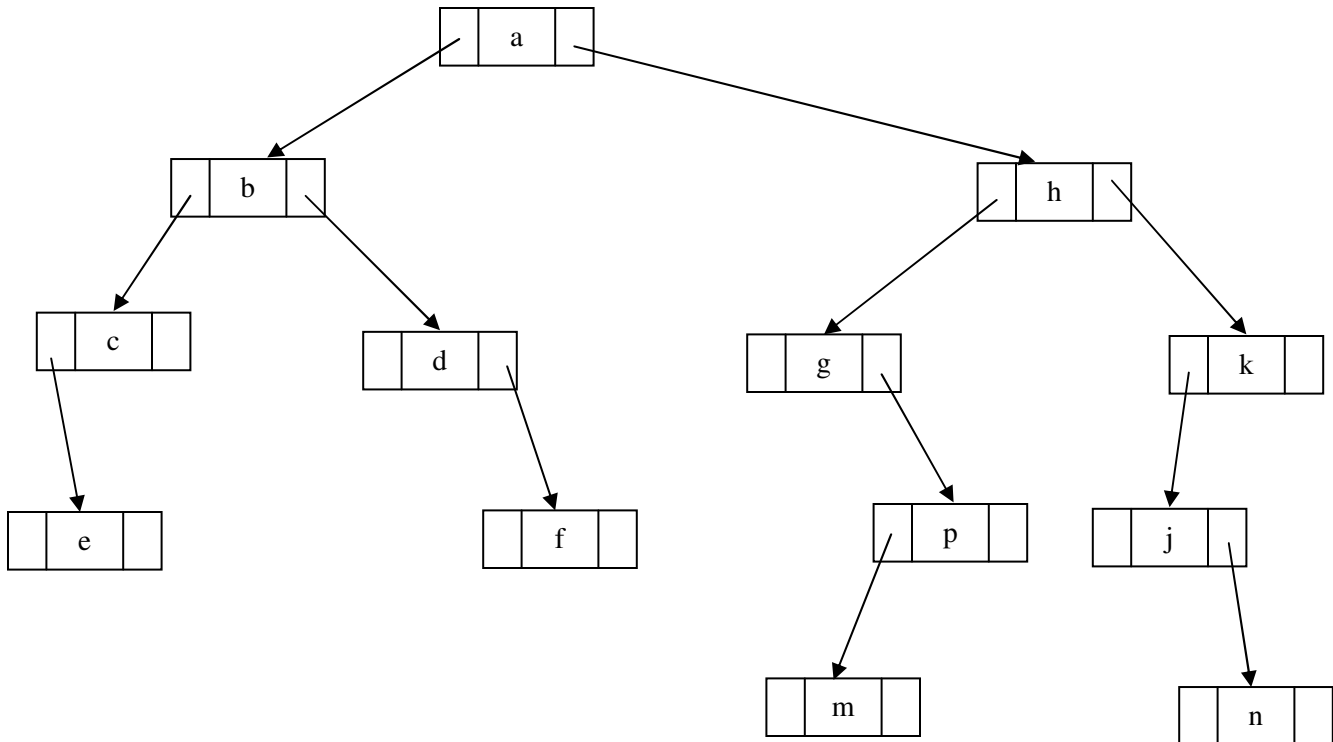
## Problem #2

A) Convert the following infix expressions into postfix and prefix.

$a * b - c * d * e * (d - f) - g$

$a * (b + c) * (d - e) - d * f$

B) Give the Preorder, Postorder, and Inorder traversals of the tree below:



**Problem #3**

Write a method that concatenates two strings (as defined below) and returns the resulting new string.

```
class string
{
    public:
        string();
        string operator+(string) const;

    private:
        char s[256]; //null terminated character array
};
```