

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

Problem 5

In C++ implement a generic **double**-linked-list class, called `List<item>`, that uses dynamic memory allocation. The list must look like the following:

$$\text{frontptr} \rightarrow X_1 \leftarrow X_2 \leftarrow \dots \leftarrow X_n \leftarrow \text{backptr}$$

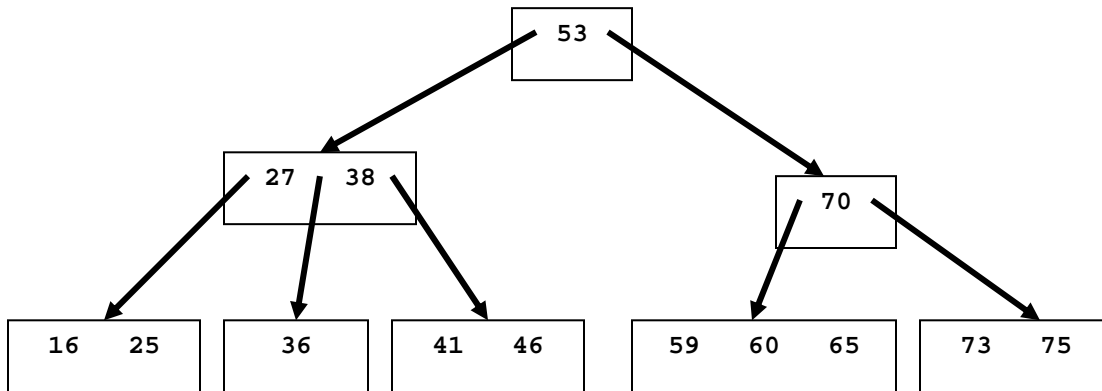
where X_1 is the first node in the list and X_n is the last node in the list. Besides `List`, you will need class or struct called `dnode`. Along with the class definition(s) you will need to implement a A) constructor, B) destructor, and C) copy constructor for `List`. Also you must implement the following member functions for `List<item>`:

- `AddBack(item)` – Adds an item to the back of the list.
- `AddFront(item)` – Adds an item to the front of the list.
- `item RemoveFront()` – removes a node from the front of a list and returns its contents.

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

Problem 6

For the following 2-3-4 tree give the corresponding red-black tree.



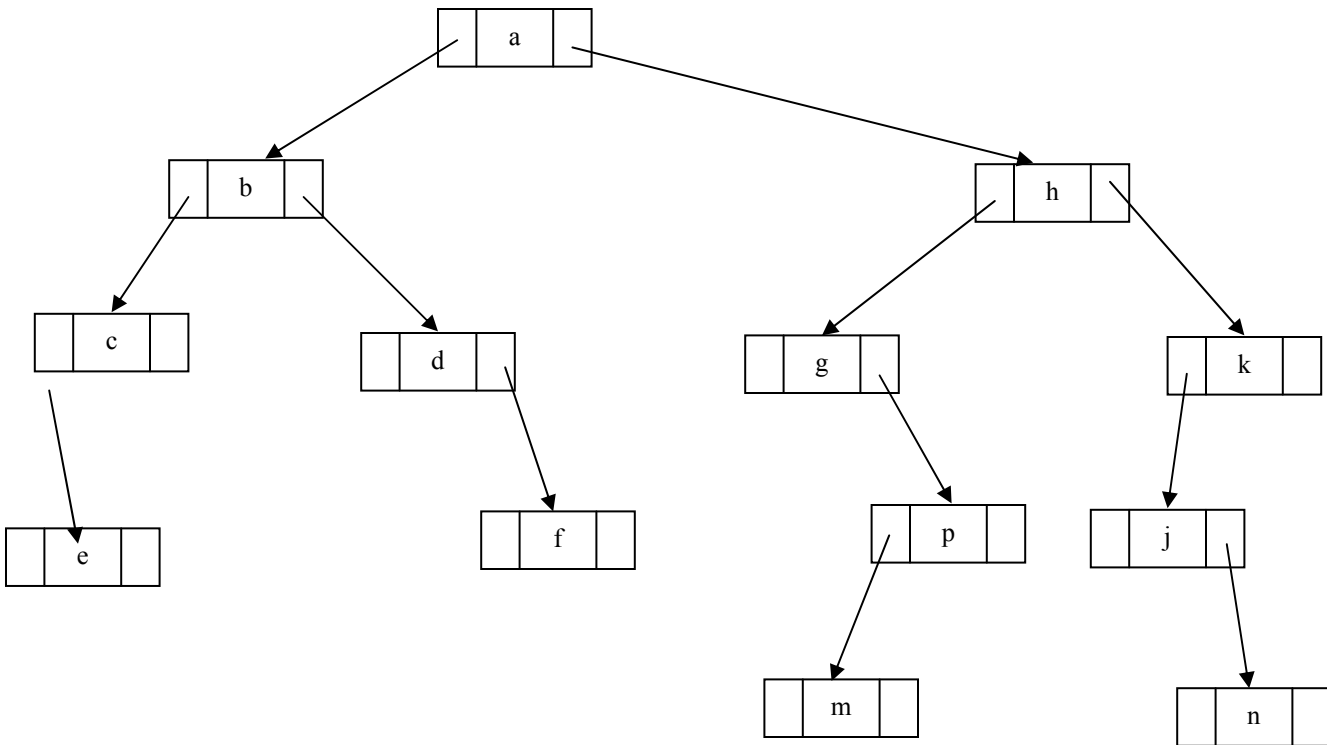
Problem 7

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 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)";
```