**Binary tree construction and traversal  
lecture 5 Tree (Due: PM:6:00, June 8, 2021)**

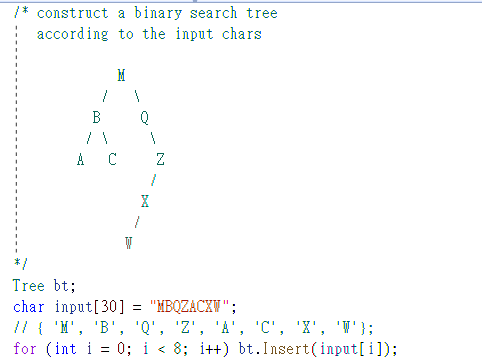
1. **Goal:**

Study how to implement binary tree construction and how to perform tree traversal.

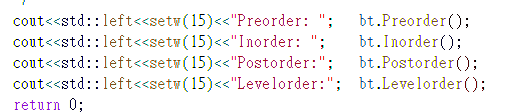
One simple program is provided for you to implement all the tree construction and traversal functions. You can design the program function with reference to the lecture notes that describe the program details. In addition, you can use binary tree

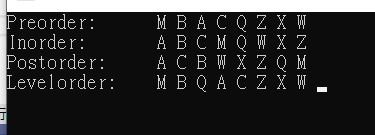
1. **Example:**

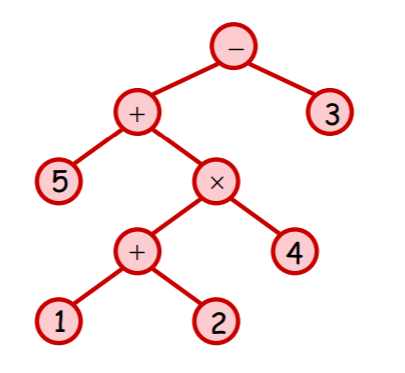
* **If the input characters are** **{ 'M', 'B', 'Q', 'Z', 'A', 'C', 'X', 'W'};**  
  Then the constructed binary search tree looks like below

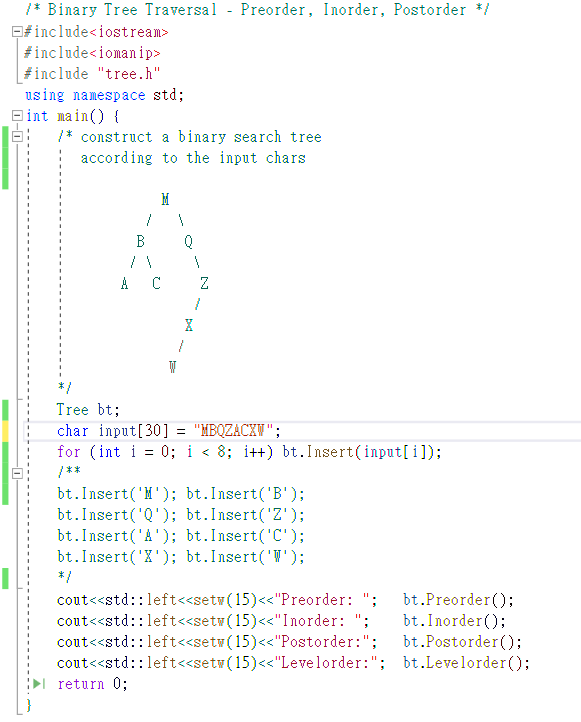


* Traversal

  
After performing pre-order, in-order, post-order and level-order traversal functions, the output would be



1. **What to do (60%)**
   * A tree.h and a main.cpp program are provided for you to start designing easily.
   * Design a binary search tree construction function.
   * Design the four binary tree traversal functions, i.e., in-order, pre-order, post-order and level order traversal functions.
   * One execution file, “tree.exe”, is provided for you to check the correctness of your program.
2. **Applications (40%)**
   * In our previous program design homework, we designed infix to postfix function. In this homework, you can write a program to construct an infix expression binary tree. For example, you can design a member function for the binarytree (bt) class and this function, bt.inorder(“5+(1+2)\*4)-3”), will construct the binary tree as shown on the right.
   * Based on this infix expression binary tree, you can design another member function bt.evaluate() that can evaluate the value of this infix expression.   
     i.e., cout << bt.**evaluate()** will output **14.**
   * You had to design this program based on the provided program template. In short, do the homework by yourself and not just download another working program available on websites.
3. **Report**Write a report to describe how do you solve this problem and design the program. Cheating will give you a big egg. “0”.

* main.cpp  
  
* tree.h  
  