Lab#8 Tree Build and Traversal

1**. Node Creation:**

```
class Node {
  private:
        int data;    Node *left;      Node *right;
       Node(int value) {data = value;    left = 0; right = 0;}
  friend class Tree;
};
```

2. ADT

Must: Preorder, Postorder, Inorder, Operator, Operand, EvalTree 등

3. Precedence Table (연산자 우선순위 테이블)

char prec[5][2] = {'^' 3, '*', 2, '/', 2, '+', 1, '-', 1};

=>

^	*	/	+	-
3	2	2	1	1

4. Main Program

1) Get mathematical expression from keyboard (ex: A+B*C)
2) Build Tree (algorithm - lecture note 참조)
3) Do Tree Traversal (Inorder, Preorder, Postorder) (printout result)
4) Use “Tree Evaluation ADT” from Lecture Note, and Compute the equation
   Ex)3+4*5= 23

- LAB Testing example

Input: 3+4*5

Output: Inorder: 3+4*5 Postorder: 345*+ Preorder: +3*45

Evaluation: 23