

Lab#8 Tree Build, Traversal, & Evaluation

1. Node Creation:

```
template <class T>
class Node {
private:
    T data;
    int priority;
    Node<T>* left;
    Node<T>* right;
public:
    Node(T value) : data (value),      priority(4), left(0), right(0) {}
    friend class Tree<T>;
};
```

2. Precedence Table (연산자 우선순위 테이블)

char prec[4][2] = { '*', 2, '/', 2, '+', 1, '-', 1};

*	/	+	-
2	2	1	1

3. Main Program

- 1) Get string expression (키보드에서 산술식 입력) (ex: 2+4*3)
 - 2) Build Tree (트리 생성 및 평가)
 - 3) Do Tree Traversal (Inorder, Preorder, Postorder) (트리 운행결과 출력)
4. Tree build 상세 알고리즘 (강의노트 참조)

5. 테스트 절차

1) Infix Expression : (8+9-2*3)

Inorder: 8+9-2*3

Postorder: 8 9 + 2 3 * -

Preorder: - + 8 9 * 2 3

- 만약 데이터가 숫자이면:

Eval Tree (알고리즘사용): **Output: 11**

2) Infix Expression: (A / B * C * D + E)

Inorder: A / B * C * D + E

Postorder: A B / C * D * E +

Inorder: + * * / A B C D E

- **Screen Output:**

```
Infix Expression1 : 8+9-2*3
InOrder : 8 + 9 - 2 * 3
PostOrder : 8 9 + 2 3 * -
PreOrder : - + 8 9 * 2 3
Evaluation : 11

Infix Expression2 : A/B*C*D+E
InOrder : A / B * C * D + E
PostOrder : A B / C * D * E +
PreOrder : + * * / A B C D E
```