

DS HW7

- Submit(제출) 방법

```
submit pem_ta hw7_ ( "_" 부분에 1분반은 a, 2분반은 b)
```

- 마감 일자: (1분반)11월 16일, (2분반)11월 13일
- 필수 제출 파일 (4개): `hw7.cpp`, `bt.h`, `학번.tex`, `학번.pdf`
 - 제출 시 실행파일을 지우고 제출합니다 (지우기 명령어 : `rm`)
- 과제 관련 문의는 ntommy11@naver.com 으로

1. hw7 디렉토리를 만들고 그 안에서 다음과 같은 `makefile` 을 만든다.

```
mkdir hw7
cd hw7
vi makefile

hw7: hw7.o
    g++ -o hw7 hw7.o
hw7.o: bt.h
```

2. 메인 함수가 있는 `hw7.cpp` 는 다음과 같다.

```
#include "bt.h"

int main()
{
    Tree<double> tree;
    double dval;

    cout << "Enter doubles:\n";

    while (cin >> dval)
        tree.Insert(dval);
    cout << endl << "Preorder traversal: "; tree.Preorder();
    cout << endl << "Inorder traversal: "; tree.Inorder();
    cout << endl << "Postorder traversal: "; tree.Postorder();
    cout << endl << "Levelorder traversal: "; tree.Levelorder();
    cout << endl;
}
```

3. 다음과 같이 동작하는 프로그램을 작성하려 한다. (출력 예시)

```
make hw7
Enter doubles:
35.3 15.7 81.5 4.5 66.7 91.2 2.3 5.2 88.2 94.5
ctrl-d
Preorder traversal: 35.3 15.7 4.5 2.3 5.2 81.5 66.7 91.2 88.2 94.5
Inorder traversal: 2.3 4.5 5.2 15.7 35.3 66.7 81.5 88.2 91.2 94.5
Postorder traversal: 2.3 5.2 4.5 15.7 66.7 88.2 94.5 91.2 81.5 35.3
Levelorder traversal: 35.3 15.7 81.5 4.5 66.7 91.2 2.3 5.2 88.2 94.5
```

4. `bt.h` 를 아래와 같이 작성하시오.

```
#ifndef TREE_H
#define TREE_H
#include <iostream>
#include <queue>
using namespace std;

template <class T>
struct Node
{
    Node(T d, Node<T> *left = 0, Node<T> *right = 0)
        : data(d), leftChild(left), rightChild(right) {}

    Node<T> *leftChild;
    T data;
    Node<T> *rightChild;
};

template <class T>
class Tree
{
public:
    Tree() { root = 0; } // empty tree
    void Insert(T &value) { Insert(root, value); }
    void Preorder() { Preorder(root); }
    void Inorder() { Inorder(root); }
    void Postorder() { Postorder(root); }

    void Levelorder();

private: // helper 함수들
    void Visit(Node<T> *);
    void Insert(Node<T> *&, T &);
    void Preorder(Node<T> *);
    void Inorder(Node<T> *);
    void Postorder(Node<T> *);
};
```

```

    Node<T> *root;
};

template <class T>
void Tree<T>::Visit(Node<T> *ptr) { cout << ptr->data << " "; }

template <class T>
void Tree<T>::Insert(Node<T> *&ptr, T &value)
{ //Insert 의 helper 함수
    if (ptr == 0)
        ptr = new Node<T>(value);
    else if (value < ptr->data)
        Insert(ptr->leftChild, value);
    else if (value > ptr->data)
        Insert(ptr->rightChild, value);
    else
        cout << endl
            << "Duplicate value " << value << " ignored\n";
}

//Preorder, Inorder, Postorder 함수를 구현하시오.....
//Levelorder(교재 p266 참조하되 STL 큐를 이용) 를 구현하시오.

#endif

```

5. **bt.h** 파일에 Preorder, Inorder, Postorder, Levelorder 함수를 구현하시오

주의사항

- * 오타가 존재 할 수 있다.
- * **Inorder** 구현시 **recursion** 을 이용하지 않고 구현하시오
- * Threaded binary tree 를 이용하여 Inorder 을 구현 할 경우 30%의 추가점수가 제공됨.
- * Levelorder 은 STL 의 queue 를 이용해 구현하시오.
- * 단 문제에 명시된 함수 이외에 **추가적인 함수 생성 불가**