

## 실습 과제 4: Binary Search Tree

실습 과제 내용:

Binary Search Tree를 직접 만들어보고, Traversal과 관련된 함수를 구현해본다.

구조체 예시:

```
struct node {  
    int value;  
    Node parent;  
    Node leftChild;  
    Node rightChild;  
}
```

구현해야 할 함수:

```
Tree createTree();  
int isEmpty(Tree T);  
Node search(Tree T, int key);  
void deleteNode(Tree T, int key);  
void insertNode(Tree T, int key);  
  
void showAll(Tree T);  
int getHeight(Tree T, int key);
```

입출력 예시 및 조건:

입력되는 key의 범위는 1 ~ 100이다.

i x: x를 value로 갖는 Node를 Tree에 삽입한다. 중복되는 값은 입력하지 않는다.

d x: Tree에서 x를 value로 갖는 Node를 삭제한다.

만약 해당 Node가 존재하지 않는다면 경고 메시지를 출력한다.

h x: x를 value로 갖는 Node의 Height를 출력한다.

만약 해당 Node가 존재하지 않는다면 경고 메시지를 출력한다.

s: Tree의 모든 노드의 value를 오름차순으로 출력한다.

e: 프로그램을 종료한다.

< 입력 >

< 출력 >

```
i 30
i 20
i 50
i 40
d 45
i 60
i 45
i 10
i 25
i 22
h 30
d 20
d 30
h 50
s
e
```

```
Not found
The height of the node (30) is 3
The height of the node (50) is 1
10 22 25 40 45 50 60
```

```
i 30
i 20
i 50
i 40
d 45
Not found
i 60
i 45
i 10
i 25
i 22
h 30
The height of the node (30) is 3
d 20
d 30
h 50
The height of the node (50) is 1
s
10 22 25 40 45 50 60
e
```

유의사항:

1. 제출 방식: Assignment4 폴더를 만들어 그 안에 Assignment4\_학번.c를 저장
2. 제출 기한: 5.11(목) 23시 59분 59초까지