

## 오픈랩 과제 #6 : Threaded Binary Tree

2021. 5. 18.

※ Threaded Binary Tree 를 구성하고 inorder의 결과를 출력한다. 메인함수와 함수의 원형, 자료구조는 아래와 같고 수정할 수 없다. 노드를 왼쪽에 삽입하는 함수인 insertLeft 함수와 빈 노드를 반환하는 함수인 getNewNode 함수는 직접 구현해야 한다.

```
typedef struct threadedTree *threadedPointer;
typedef struct threadedTree {
    short int leftThread;
    threadedPointer leftChild;
    char data;
    threadedPointer rightChild;
    short int rightThread;
} threadedTree;
threadedPointer insucc(threadedPointer tree);
void tinorder(threadedPointer tree);
void insertRight(threadedPointer s, threadedPointer r);
void insertLeft(threadedPointer s, threadedPointer l);
threadedPointer getNewNode(char data);
int main() {
    /* image 5.23 */
    threadedPointer root = getNewNode('-');
    insertLeft(root, getNewNode('A'));
    insertLeft(root->leftChild, getNewNode('B'));
    insertLeft(root->leftChild->leftChild, getNewNode('D'));
    insertLeft(root->leftChild->leftChild->leftChild,
getNewNode('H'));
    insertRight(root->leftChild->leftChild->leftChild,
getNewNode('I'));
    insertRight(root->leftChild->leftChild, getNewNode('E'));
    insertRight(root->leftChild, getNewNode('C'));
    insertLeft(root->leftChild->rightChild, getNewNode('F'));
    insertRight(root->leftChild->rightChild, getNewNode('G'));

    tinorder(root);
    return 0;
}
```

(1) 입력 : X

(2) 출력 :

- 출력은 표준 출력으로 출력한다. 첫째줄에 inorder로 탐색한 결과를 출력한다

※ 출력 결과

H	D	I	B	E	A	F	C	G
---	---	---	---	---	---	---	---	---

(4) 과제

- 과제 실행화면, C소스코드(실행파일은 제외 시켜주길 바랍니다.)를 과제 게시판에 업로드

\* 멘토의 메일로 전송하거나 지각제출일 경우 0점처리 되니 꼭 과제 게시판에 업로드하기 바랍니다.

제출기한 : ~ 5월 20일 (목) 23:59:59 까지