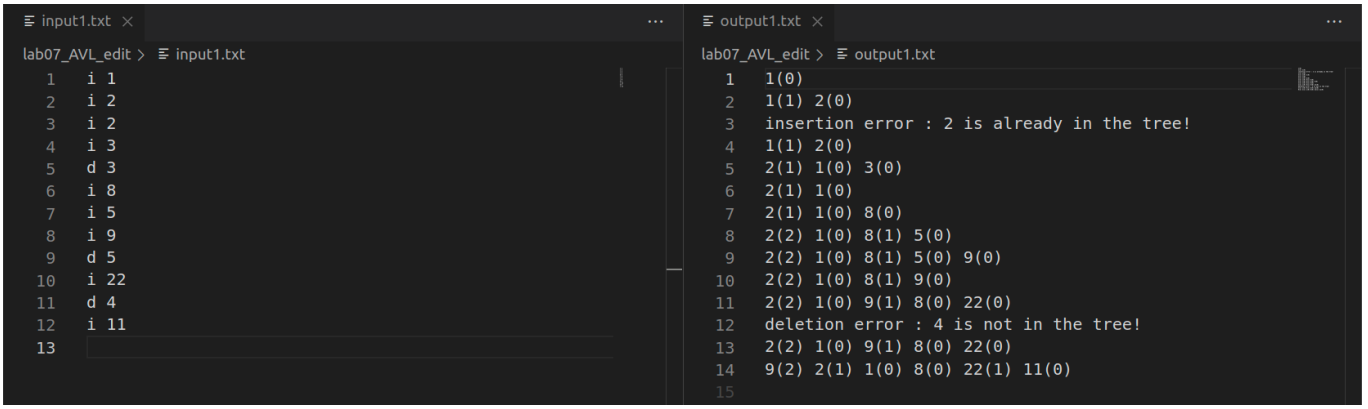
2023 Data Structure (CSE2010 12282)

Lab 06 (4/13) AVL Tree

- arranged by TA Sehyun Cha

Due : ~2023.04.19.(Wed) 23:59, Late Submission : ~2023.04.20.(Thu) 23:59

- **AVL Tree Implementation - Insertion, Deletion, PrintPreorder.**
- 아래의 사진과 같이 **[input].txt** 파일의 커맨드를 입력받아 **AVL Tree Insertion, Deletion** 기능을 수행하고, 각 커맨드가 수행될 때마다 상황에 맞은 메시지와 **Preorder Traversal**의 결과를 **[output].txt** 파일에 출력하여 저장한다.
- **Implement the function of creating and editing AVL tree by receiving the [input].txt file as input as below and the output results of preorder traversals stored in the [output].txt file.**



input1.txt

```
i 1
i 2
i 2
i 3
d 3
i 8
i 5
i 9
d 5
i 22
d 4
i 11
```

output1.txt

```
1(0)
1(1) 2(0)
insertion error : 2 is already in the tree!
1(1) 2(0)
2(1) 1(0) 3(0)
2(1) 1(0)
2(1) 1(0) 8(0)
2(2) 1(0) 8(1) 5(0)
2(2) 1(0) 8(1) 5(0) 9(0)
2(2) 1(0) 8(1) 9(0)
2(2) 1(0) 9(1) 8(0) 22(0)
deletion error : 4 is not in the tree!
2(2) 1(0) 9(1) 8(0) 22(0)
9(2) 2(1) 1(0) 8(0) 22(1) 11(0)
```

- <input> : 각 line 마다 Command 가 주어짐.(**insert, delete**)
- <output> :
  각 **Command** 마다 수행후의 **AVL Tree**의 상태를 출력. **(Tree** 상태 출력시 **preorder traversal** 결과를 출력, 출력하는 **Key** 값마다 해당 노드의 **height**를 함께 출력**)**
  Tree에 이미 존재하는 **Key** 값을 **Insert** 하는 경우, 에러 메시지를 출력.
  Tree에 존재하지 않는 **Key** 값을 **Delete** 하는 경우, 에러 메시지를 출력.
  (에러메시지를 출력하는 경우에도 **Tree** 상태는 항상 출력.**)**
- <input> : commands(insert or delete) are given line by line.
- <output> :
  State of AVL tree after executing each command should be printed.(preorder traversal results with height information of each node.)
  Print error messages when inserting the key value that already exists in the tree or deleting the key value that does not exist in the tree.
  (State of the tree should be printed out even when the error message is printed out.)

<Structure & Function Format>

Structure

```
struct AVLNode;
typedef struct AVLNode *Position;
typedef struct AVLNode *AVLTree;
typedef int ElementType;

typedef struct AVLNode{
    ElementType element;
    AVLTree left, right;
    int height;
}AVLNode;
```

Function

```
AVLTree Insert(ElementType X, AVLTree T);
AVLTree Delete(ElementType X, AVLTree T);
Position SingleRotateWithLeft(Position node);
Position SingleRotateWithRight(Position node);
Position DoubleRotateWithLeft(Position node);
Position DoubleRotateWithRight(Position node);
void PrintPreorder(AVLTree T);
void DeleteTree(AVLTree T);
```

- 위 사진과 같은 Struct 구조체를 사용하셔야 합니다.
- 위 사진과 같은 함수들을 형식에 맞게 구현해주시면 됩니다.
- Struct format above should be used for implementation
- Functions should be implemented in appropriate format as above.

<Insert>

- AVL Tree에 입력받은 key값을 삽입합니다.
- Tree에 이미 존재하는 key값을 Insert 하는 경우, 에러메시지는 **"insertion error : key값 is already in the tree!"** 의 형식으로 출력해주셔야 합니다.
- Insert given key into AVL Tree
- When inserting the key value that already exists in the tree, the error message format is **"insertion error : keyValue is already in the tree!"**

<Delete>

- AVL Tree에 입력받은 key값을 삭제합니다.
- Tree에 존재하지 않는 key값을 Delete 하는 경우, 에러메시지는 **"deletion error : key값 is not in the tree!"** 의 형식으로 출력해주셔야 합니다.
- Delete given key into AVL Tree
- When deleting the key value that does not exist in the tree, the error message format is **"deletion error : keyValue is not in the tree!"**

<PrintPreorder>

- 출력시, 위 사진의 예시와 같은 형식으로 출력해주시면 됩니다. 모든 공백은 띄어쓰기 한칸입니다. 모든 출력 메시지의 알파벳은 소문자만 사용하여 출력합니다.
  -> preorder 순회 결과 출력시 마지막 ")" 뒤에 줄바꿈(\n).)뒤에 공백있음.
  -> 에러메시지 출력 상황의 경우, 에러메시지 출력후 줄바꿈(\n)후 preorder 순회 결과 출력.
  -> 출력의 마지막 줄인 경우 줄바꿈 후 EOF.
- All the messages must be printed out according to the appropriate format as shown in the example above. All spaces are one space. Only lowercase letters should be used for the alphabets in output messages.
- -> newline(\n) after each preorder traversal (after last ")"). there is space after)
- -> when printing an error message, newline(\n) after the error message and print preorder traversal result in the next line.

- -> newline(\n) after the last output message.

<File Name Format>

- [StudentID].c ex) 20XXXXXXXX.c

<Execution>

- gcc 20XXXXXXXX.c -o 20XXXXXXXX
- ./20XXXXXXXX [input_file_name] [output_file_name]
- !!! 꼭 제공되는 testcase로 실행시켜보시기 바랍니다.!!!!!,
- !!! Run your solution code with the provided test case above and check whether it works properly !!!

<Issue>

- 코드 작성시 주석을 적어주시기 바랍니다. 주석이 없는경우 Cheating으로 간주될 수 있습니다.
- 제공된 testcase는 채점 case에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
- 파일 입출력은 *argv[]* 를 사용하여 구현해주시기 바랍니다.
- 제출 마감 시간 이전의 가장 최신 버전의 commit을 기준으로 채점할 예정입니다.
- 제출 파일과, 폴더 naming 은 꼭 지정된 형식으로 해주셔야 합니다.
- Please write down the detailed comments when writing the code. If there is no comment, it might be considered cheating.
- Provided test case is included in the test case for grading. Please check to see if it makes a proper result.
- Do not use a fixed file name when inputting and outputting files, but implement it using argv[] as in skeleton code.
- Scoring will be based on the latest version of commit before the deadline.
- The names of the .c file and directory should be named in proper format.
- 에러 메시지 출력후 프로그램이 종료되면 안됩니다.**(**남아있는 커맨드를 모두 수행해야 합니다.**)**
- 각 커맨드가 수행될 때마다 Tree의 상태를 preorder traversal 순서로 출력해주셔야 합니다.
  출력형태는 위 사진과 같은 형식으로 출력해주시면 됩니다.
  -> **"key값(height) key값(height) key값(height) key값(height) ...."**
- 에러메시지를 출력해야하는 경우 위 예시사진처럼, 에러메시지를 먼저 출력후, 줄바꿈 후에 Tree의 상태를 preorder traversal 순서로 출력하셔야 합니다.
- **Program must not be terminated when printing out the error messages.(All the remaining commands should be executed properly.)**
- For each command, the state of the tree should be printed in preorder traversal order as the format below.
  -> **"keyValue(height) keyValue(height) keyValue(height) keyValue(height) ...."**
- when printing an error message, as an example above, the error message must be printed first and the preorder traversal result should be printed in the next line.

<Directory Format>

- 아래와 같이 git 프로젝트 폴더에 “lab06” 폴더 생성후, "lab06" 폴더 안에 "20XXXXXXXX.c" 파일을 위치시키시면 됩니다.
- After creating the "lab06" directory in the git-project-directory as below, place the "20XXXXXXXXX.c" file in the "lab06" directory.

>>>

2023_CSE2010_20XXXXXXXX/ (GitLab project directory)

---...

---lab04/

-------2022XXXXXX.c

——--...

---lab05/

-------2022XXXXXX.c

——--...

---lab06/

-------2020XXXXXX.c

>> QnA : 2023ds12282@gmail.com