**4th Assignment**

2013011491

컴퓨터 전공

안찬영

TreeNode \* createTreeNode(void)

{TreeNode\*tmp = (TreeNode\*)malloc(sizeof(TreeNode));

tmp->left = NULL;

tmp->right = NULL;

return tmp;}

**먼저 노드 생성 함수 입니다. 동적 할당으로 노드를 생성 한 뒤, 왼쪽,오른쪽 subtree들을 NULL로 초기화 합니다.**

void setData(TreeNode \* node, element data)

{node->data = data; }

**노드에 데이터를 저장 하는 함수입니다**.

element getData(TreeNode \* node)

{return node->data;}

**노드에 저장되어 있는 데이터를 출력하는 함수입니다**.

void makeLeftSubTree(TreeNode \* main, TreeNode \* sub)

{if(main->left!=NULL)

exit(1);

else

main->left=sub;}

**main노드 왼쪽 subtree에 sub노드를 생성하는 함수입니다. 이미 왼쪽 subtree가 있을 경우에는 함수를 종료 시킵니다.**

void makeRightSubTree(TreeNode \* main, TreeNode \* sub)

{if(main->right!=NULL)

exit(1);

else

main->right=sub;}

**위 makeLeftSubTree함수와 설명이 유사하므로 생략하겠습니다.**

TreeNode \* getLeftSubTree(TreeNode \* node)

{return node->left;}

**노드의 왼쪽 subtree를 반환하는 함수입니다,**

TreeNode \* getRightSubTree(TreeNode \* node)

{return node->right;}

**노드의 오른쪽 subtree를 반환하는 함수입니다,**

void printInorder(TreeNode \*root){

if(root==NULL)

return;

printInorder(getLeftSubTree(root));

printf("%d ",getData(root));

printInorder(getRightSubTree(root)); }

**중위순회 출력 함수입니다. 먼저 재귀 탈출 조건을 명시합니다. NULL에 도달할 경우 반환하는 조건입니다. 그 다음 중위 순회 이므로 왼쪽 subtree에 이동 후 중위 순회 합니다. 왼쪽 subtree에 순회가 끝나면 node의 데이터를 출력한 뒤 오른쪽 subtree로 이동 후 다시 중위 순회합니다**.

void printPreorder(TreeNode \*root){

if(root==NULL)

return;

printf("%d ",getData(root));

printPreorder(getLeftSubTree(root));

printPreorder(getRightSubTree(root));

}

**위 중위 순회와 설명이 유사합니다. 노드의 데이터를 먼저 출력하므로 데이터를 출력한 뒤, 왼쪽으로 가서 순회, 그 다음 오른쪽으로 가서 순회 합니다.**

void printPostorder(TreeNode \*root){

if(root==NULL)

return;

printPostorder(getLeftSubTree(root));

printPostorder(getRightSubTree(root));

printf("%d ",getData(root)); }

**위 순회 함수들과 설명이 유사합니다. 노드의 데이터를 마지막에 읽으므로, 왼쪽에서 먼저 순회한 뒤, 오른쪽 순회 후, 데이터를 출력합니다**.

int main(int argc,char\*argv[]){

FILE \*input;

input = fopen(argv[1], "r");

char command;

TreeNode\*t1=createTreeNode();

TreeNode\*t2=createTreeNode();

TreeNode\*t3=createTreeNode();

TreeNode\*t4=createTreeNode();

TreeNode\*t5=createTreeNode();

TreeNode\*t6=createTreeNode();

TreeNode\*t7=createTreeNode();

TreeNode\*t8=createTreeNode();

TreeNode\*t9=createTreeNode();

TreeNode\*t10=createTreeNode();

setData(t1,54);

setData(t2,27);

setData(t3,89);

setData(t4,13);

setData(t5,44);

setData(t6,71);

setData(t7,92);

setData(t8,1);

setData(t9,37);

setData(t10,64);

makeLeftSubTree(t1,t2);

makeRightSubTree(t1,t3);

makeLeftSubTree(t2,t4);

makeRightSubTree(t2,t5);

makeLeftSubTree(t4,t8);

makeLeftSubTree(t5,t9);

makeLeftSubTree(t3,t6);

makeRightSubTree(t3,t7);

makeLeftSubTree(t6,t10);

while(1){

command = fgetc(input);

if(feof(input)) break;//입력 파일 다 읽을 시 반복문 탈출

switch(command) {

case 'i': printf("Inorder: "); printInorder(t1); printf("\n");

break;

case 'r': printf("Preorder: "); printPreorder(t1); printf("\n");

break;

case 'o': printf("Postorder: "); printPostorder(t1); printf("\n");

break;

default:

break;}}

fclose(input);//파일 종료

return 0;}

**노드들을 생성한 뒤 데이터를 설정해주고, 각각위치에 맞게 subtree들을 설정해서 이진 트리를 만들어 줍니다. 그 다음 파일로부터 입력 받는 커맨드에 따라 중위순회, 전위순회, 후위순회를 실시합니다.**