Project2: Constructing a binary tree and tree traversal

• Content : We read records in the file and store them in the binary tree. After building a tree, we preform tree traversals of three methods: inorder, preorder, and lever order.

• Method:

(1) In each line of the file, the name and gpa of a student are stored. We use name as the key. To read a line, we can use the function fscanf.

(2) The nodes of the tree are prepared by dynamic allocation. The parent node use pointers to point to its children. We do not use an array for the tree. The following definition is used.

typedef struct node \* typeNodePtr;

typedef struct node {

char name[30];

double gpa ;

treePointer leftChild, rightChild;

} typeNode ;

(3) We use a global variable ROOT to point to the root node of the tree. It has the value NULL initially. The first line of the file has line number 1. If the line we read has line number n, its node is attached to the tree so that its node number is n in the binary tree. To create a node we use the function malloc. The final result of the tree will be a complete binary tree. However, we do not use an array to store the tree.

(4) For final tree, perform inorder traversal. When a node is visited, its content (name and gpa) is printed on the screen. In themain function, we use the call: inorder(ROOT);

(5) Perform preorder traversal by calling: preorder(ROOT);

(6) Perform levelorder traversal by calling: leverorder(ROOT);

• The content of the file: (We create a text file named Studentdata.txt and read from this file.)

Park 3.4

Jung 3.8

Lee 4.1

Joo 3.1

Seo 4.2

Ban 2.9

Nam 3.0

Sun 2.5

Min 3.6

Cho 3.2

Tho 2.7

Jin 3.7

For submission: The project directory is zipped. Please prepare a file containing the captured image of the screen showing the result of the run of the program. (with file name capture.jpg). Put the file in the project directory before doing zip.