**CSCI 520L.01L Lab**

**Programming Assignment #8**

To be done in the lab

(Friday, Oct. 24, JOUR 102)

Upload your work to Drop Box for Week 8 in eCollege

Write a C++ program that creates a tree from given input, and prints the current tree using inorder traversal. Implement the following steps:

The following is how we define tree nodes:

struct node

{ int item; node \*l, \*r;

node(int x)

{ item = x; l = 0; r = 0; }

};

typedef node\* link;

The trees for this assignment have the following properties:

1. Each node has zero, one, or two children (left child and right child);
2. Each node has a unique key. That is, the item in each node is different from items in all other nodes.
3. Each input item is also unique.

You may find the following function useful. This function visits and prints nodes in a tree in in-order traversal order starting at the given root h:

void inordertraverse(link h)

{

if (h == 0) return;

inordertraverse(h->l);

cout << h->item;

inordertraverse(h->r);

}

Your program will receive the following input and do the following for each input:

C <int r> : Create the root node with item r. If the tree is not empty (i.e. the root exists), return an error

L <int p> <int l>: Find the node with item p. Let u be that node. Create a new node v with item l. Make node v node u’s left child. If node u has already a left child, return an error

R <int p> <int r>: Find the node with item p. Let u be that node. Create a new node v with item r. Make node v node u’s right child. If node u has already a right child, return an error

P: Print the items in nodes in the entire tree in in-order traversal manner

S <int p>: Find the node with item p. Let u be that node. Print the items in nodes in the subtree rooted at u in in-order traversal manner

See the example dialogue below:

C 5

Root node with item 5 has been created

C 10

Error: Tree is not empty

L 5 3

Node with item 3 has been added

R 5 4

Node with item 4 has been added

P

in-order traversal of the entire tree: 3 5 4

S 4

in-order traversal of the asked subtree: 4

L 4 7

R 7 8

R 3 9

L 9 11

R 4 12

L 7 13

S 3

in-order traversal of the asked subtree: 3 11 9

S 4

in-order traversal of the asked subtree: 13 7 8 4 12

L 8 14

R 12 15

R 8 16

L 13 17

P

in-order traversal of the entire tree: 3 11 9 5 17 13 7 14 8 16 4 12 15