This is a two part project. In the first part you will construct a binary search tree and node class, with some basic functions. In the second part, you will expand the functionality of the binary search tree. Test your program with the code files below:

<https://markbowman.org/231/Program05.zip>

You may not change the Main.cpp file. You will need to create the Tree.h, Tree.cpp, Node.h, and Node.cpp files.

Tree Class

The binary search tree class should have the following data elements:

* Root Pointer to the root node

The binary search tree class should have the following functions:

* Constructor Initialize the empty tree
* Destructor Delete all the nodes in the tree
* Insert Insert a new value into the tree
* Show Display all the values
* Find Return true/false if the string is found in the tree

Node Class

The node class should have the following data elements:

* Value Stored value
* Left, Right Pointers to children nodes

The node class should have the following functions:

* Constructor Create an empty node
* Destructor Delete all children of this node
* Put Output the value
* LMR Output the current node and all children values in LMR order

The value and children pointers should be private to the node class. It may help to declare the tree class a friend to the node class.

What to Hand In

Hand in a copy of each code file (don’t include Main.cpp), and runs with both input files to demonstrate that your program and functions work. Sample runs are shown on the next page.

Sample Runs

|  |  |  |
| --- | --- | --- |
| Enter file name: ***People.txt***  LMR Order  --------------  Alisa  Ann  Carlos  David  Frank  John  Kathy  Lailee  Lisa  Mark  Michael  Richard  Rochelle  Steve  Susan  Enter value to find: ***Frank***  Find: Frank found! |  | Enter file name: ***People.txt***  LMR Order  --------------  Alisa  Ann  Carlos  David  Frank  John  Kathy  Lailee  Lisa  Mark  Michael  Richard  Rochelle  Steve  Susan  Enter value to find: ***Carlos***  Find: Carlos found! |
|  |  |  |
| Enter file name: ***Numbers.txt***  LMR Order  --------------  Eight  Five  Four  Nine  One  Seven  Six  Ten  Three  Two  Enter value to find: ***Two***  Find: Two found! |  | Enter file name: ***Numbers.txt***  LMR Order  --------------  Eight  Five  Four  Nine  One  Seven  Six  Ten  Three  Two  Enter value to find: ***Eleventeen***  Find: Eleventeen not found |