===Lab Info===

* 60 points

* Due 11:59pm, Wednesday 9/9 for Monday and Wednesday lab.

==Assignment==

This lab focuses on operations on a binary tree. The file data.txt contains the information about the tree in a level order format. That is, the first line contains the value at the root; the second line contains two numbers corresponding to the left and right children of the root, respectively; each remaining line contains information about the children of the nodes on the previous level. A zero in the input indicates that there is no child so the pointer to its position is nil.

You may assume that all values in the tree are distinct and positive. Once you have built the tree, do each of the following:

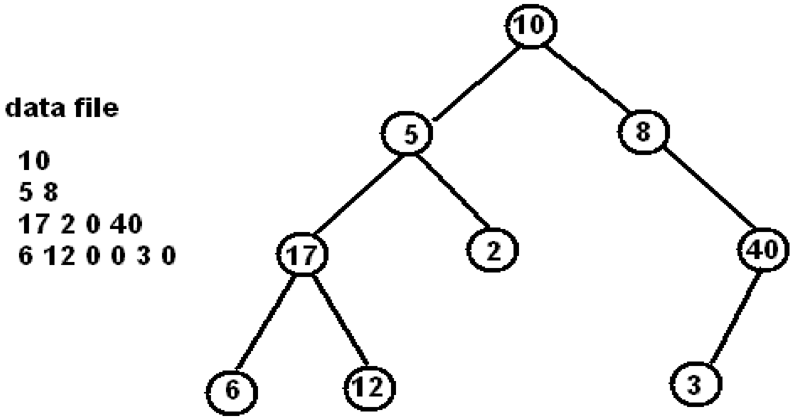
1. Using an inorder traversal, write out the values in the tree.

2. Write a code to find the largest element in the tree and print out that value.

3. Write a code that returns the number of leaves in the tree. Again print out that value.

Although you can find out the information in 2 and 3 from inspecting the data file, you won't get credit for finding the answers in that way. Functions that you write must be used to obtain the answers.

==Output==



data file

```
10
5 8
17 2 0 40
6 12 0 0 3 0
```

Output:
Inorder: 6 17 12 5 2 10 8 3 40
Largest element: 40

Number of leaves: 4

==List==

The List should implement an appropriate constructor and destructor. The rest of the methods should be implemented as follows:

*Build()   Build a binary tree from data.txt.

*Largest() Find the largest element in the Binary tree.

*Leaves() Find the number of leaves in the Binary tree.

*print() should print out all the elements of the tree separated by a space using inorder traversal.

===Files===

* Files to include in folder:

** all source files

** a functioning makefile

** data.txt

* Folder name: Lastname_Lab2

* Compressed file name: Lastname_Lab2.zip (or .rar or .tar.gz)

* Executable name: lab2