COP3530 – Data Structures and Algorithms I - Fall 2016

Programming Project#4

*** Due 11/23/2016, Wednesday, 05:00 p.m.**

Instructions:

This is a programming project to be done and submitted individually. Please make sure your solutions are original, in your own words, expressing your own ideas. Any copying from others' home-works, text books or online sources will be penalized. ***I'll be checking the code for plagiarism, so, be advised****.*

IMPORTANT: You can use any IDE for creating your programs. Create the programs and save them. Compile and run the programs. Save the screenshot/s of the successfully run programs and submit the screenshot/s with the corresponding source code files on the Dropbox. Do not submit any executable (.exe) files.

Programs should be well-commented to make their understanding easy. Variable-names should be so chosen that it is easy to visualize what is going on, inside the program. Points would be taken off for poor documentation and readability.

It is highly recommended that students keep a soft copy of the assignment files with them. Any queries can be mailed to amishra@uwf.edu.

Any exceptions to this should be pre-approved by the Instructor; late submissions (only the ones pre-approved by the instructor before the deadline) are liable to be penalized.

Previous instructions about coding submissions stay as they are.

WHAT TO TURN IN:

Upload through the "Assignments" page on Dropbox, all the files in your submission. Submit **each** of your C programs: one program for each of the above exercises. In addition, submit **all relevant screen captures of your successfully running programs** as jpeg files using the same name (for example, the screen captures for successfully running *sumOfCubes.c* should be named as *SumOfCubes1.jpg*, *SumOfCubes2.jpg* – if more than one relevant screen captures are included). Do not submit the object files or executables.

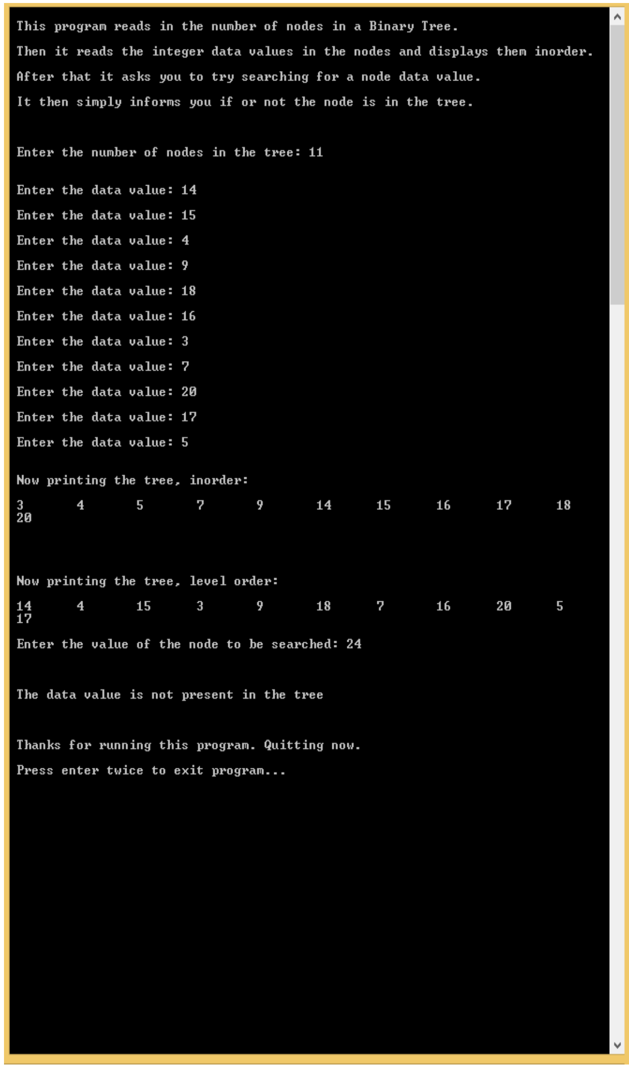
In order for your submission to be counted as on time, it should be submitted before the indicated deadline on the due date unless otherwise specified. The penalty for late submissions is 5% for each day late, and late submissions need to be preapproved by the instructor (they should NOT be automatically assumed as granted). The very latest that you may submit a late lab is one week after the submission was due.

Write a complete C program with as many functions as required to achieve the following.

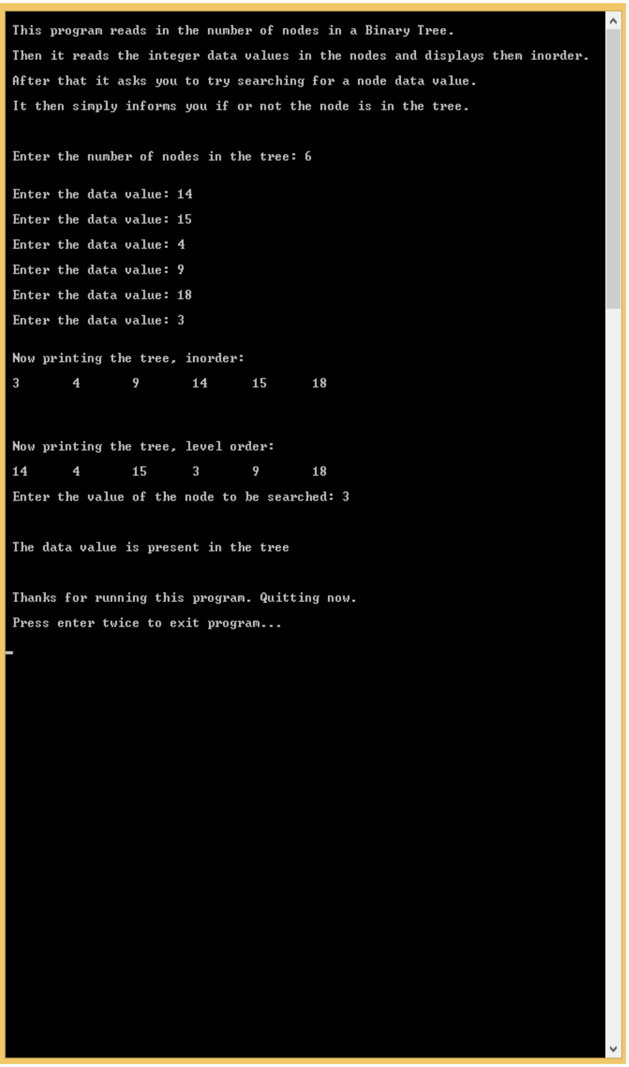
1. Create a Binary Search Tree. Each node of this tree is supposed to have an integer data item. To create your tree, you should first explain how your program works, then let the user specify the nodes of the BST. For this, you ask the number of tree nodes that the user wishes to create, and then let the user specify the data values for the BST nodes. (2 points)
2. Based on the search key integer data provided by the user, you will need to check your tree and report if the search key was found in your tree. (5 points)
3. The visualization and check on whether the tree has been created correctly – are provided by the outputs of the various traversals. You should generate the outputs of the ***inorder*** (5 points) and ***level order*** (8 points) traversals to provide this visualization.

   Important: Please understand that the points in parts 1 and 2 would be admissible only if you have successfully created the tree, which will be evident only if both your traversals are correct. This also means that there are no partial points to be taken from part of the project submitted.

   Make your program output to look like the following sample run (try out different numbers of tree nodes):



```
This program reads in the number of nodes in a Binary Tree.
Then it reads the integer data values in the nodes and displays them inorder.
After that it asks you to try searching for a node data value.
It then simply informs you if or not the node is in the tree.


Enter the number of nodes in the tree: 11

Enter the data value: 14
Enter the data value: 15
Enter the data value: 4
Enter the data value: 9
Enter the data value: 18
Enter the data value: 16
Enter the data value: 3
Enter the data value: 7
Enter the data value: 20
Enter the data value: 17
Enter the data value: 5

Now printing the tree, inorder:
3       4       5       7       9       14      15      16      17      18
20


Now printing the tree, level order:
14      4       15      3       9       18      7       16      20      5
17
Enter the value of the node to be searched: 24


The data value is not present in the tree


Thanks for running this program. Quitting now.
Press enter twice to exit program...
```



```
This program reads in the number of nodes in a Binary Tree.
Then it reads the integer data values in the nodes and displays them inorder.
After that it asks you to try searching for a node data value.
It then simply informs you if or not the node is in the tree.


Enter the number of nodes in the tree: 6

Enter the data value: 14
Enter the data value: 15
Enter the data value: 4
Enter the data value: 9
Enter the data value: 18
Enter the data value: 3

Now printing the tree, inorder:
3       4       9       14      15      18


Now printing the tree, level order:
14      4       15      3       9       18
Enter the value of the node to be searched: 3


The data value is present in the tree


Thanks for running this program. Quitting now.
Press enter twice to exit program...
_
```

***