CS 271 and 462

PA 6 - Programming Assignment 6

Overview of This Assignment

4 Files:

- 2 C source files
- 1 header file
- 1 makefile

You must upload the files to a CS Linux host, make, then run the executable on Linux. Once you have finished and tested the programs, you will upload them to the PA 6 assignment in Canvas.

C Programming Skills That You Are Practicing in This Assignment

1. Become proficient with using structures.
2. Master the process of using pointers, including the use of * for declaring and dereferencing and the & operator.
3. Become proficient with dynamic memory allocation.
4. Develop an understanding of the binary search tree data structure.

3 files are provided to you

1) binaryTree.h
2) binaryTree.c
3) pa6.c

Use these files as a starting point.

makefile - do this first!

- Include an "all" target that builds an executable named pa6 (no extension).
- Write separate targets for pa6, pa6.o, and binaryTree.o.
- Include a "clean" target that removes all object files (all files ending with .o).

Make sure you test the clean target.

Required components of this assignment:

1. Documentation and Style
2. You must add 3 function prototypes to the .h file.
3. You must add 3 function definitions to the binaryTree.c file.
4. You must extend the body of the main function in pa6.c so that it will thoroughly test the 3 functions.
5. Meaningful messages must be displayed with all output.
6. Programs must compile. If you submit any program or makefile that contains a syntax error, the grader will enter a score of 0 for that program or makefile.

Function 1

int min (TreeNodePtr treePtr);

The function should find and return the minimum value in the tree. If there are no nodes in the tree, the function should return 0.

Function 2

int max (TreeNodePtr treePtr);

The function should find and return the maximum value in the tree. If there are no nodes in the tree, the function should return 0.

Function 3

int sum (TreeNodePtr treePtr);

The function should find and return the sum of all values in the tree. If there are no nodes in the tree, the function should return 0.

Compile, test, and debug your program as needed.

Submit 4 files. Do not zip or tar the files.

- pa6.c
- binaryTree.h
- binaryTree.c
- makefile (You will need to copy this to a file named makefile.c so you can submit it.)