 # ECE 361 Homework #3 -- Binary tree ADT, reading from files, the make utility

Introduction

A binary tree ADT for MLS teams. Includes example of make file utility.

The first file MLSapp_Helpers.c which uses two functions. One for parsing items into a stack and another to print a teams information.

The seccond file MLStree.c contains all ADT items to build a binary search tree with nodes. Functions createtree, insert nodes, recursive inorder printing, buildmlstree, print final standings and conference winners with winner announcement.

The third file test2_MLStree.c uses the first two files to build an MLS binary search tree and show the standings as well as the winner.

The MLSTree folder is for the codeblocks environment for GUI based debugging purposes.

Future improvement of the code

1.) More debug print statements. Deleted quite a few due to minimizing text.

Also note the cygwin1.dll package was kept in the folder as the executable requires this dll to run except for the terminal interface.

//Deliverables Checklist

*** DONE -** Ensure that you implement the MLS app using Binary Search Tree ADT with functions like createTree() and insertNode().

*** DONE -** Ensure that your code parses the MLS2020.txt file, preferably inside the buildMLSTree function.

*** DONE without keeping track(note its faster to parse(hence the reason for creating a tree as well) two complete trees afterwards than do single comparisons while building a tree…) -** Ensure that you implement a buildMLSTree function that parses the data, inserts nodes to appropriate tree, and keeps track of the statistics to determine the shield winner.

*** DONE -** Include a Makefile to test your code.

*** DONE -** Ensure that correct results are printed similar to the sample results provided (it should look clean like a table).

*** DONE -** Please submit a transcript of your output in <username_transcript>.txt file.

*** DONE -** If you have used an online resource excluding than the sample_code provided by Professor, please cite your source link as comments.

*** DONE -** Ensure that your code does not print unnecessary debug messages using _VERBOSE_ debug macro in your Makefile.

*** DONE** - Please add meaningful comments to your code and/or a README.pdf file to explain your work.

##Deliverables Stated First

This assignment involves inserting nodes into, and traversing binary trees.

The application reads a file containing information on the 2020 MLS season and

...inserts the information into two binary trees, one for the Eastern Conference

...teams and another for the Western Conference teams.

The main program of the application traverses the tree(s) and displays "the Table"

...(the standings) for each conference. The application also uses the data from both

...trees to calculate the Shield winner...the team with the best record (in points-per=game)

...for all of the MLS teams. Your task is to design and implement a binary tree ADT and then

...to use that tree in the application. The details are in the Homework #3 write-up and

...in all of the starter code that I provided. The sample code will be helpful in understanding

...and crafting your ADT. You are also tasked with providing and using a makefile for building

...your application. The example that I reviewed in class today and posted (my makefile from HW #3 in Fall 2019) will help you with that.

This assignment should be completed by 10:00 PM on Sat 14-Nov-2020.

We are using GitHub Classroom for this assignment so make your final commit to your GitHub private repository for the assignment before the deadline.

Also, please upload a .zip file of your final repository to your Homework #3 dropbox in D2L.

C Source code for your programming solutions should be text files (not .docx, etc.) that have a .c extension.

Header files should be text files that have a .h extension.

Your transcripts (logs) should be submitted as text files (.log) by either redirecting the output from your shell to a file or by using the BASH tee command

to send output to both the console and a file (ex: $ echo “hello world” |tee helloworld.log). You may also copy/paste a transcript into a .txt file and submit that.

The transcript should identify yourself, show your make file in operation (make clean, make -n), show the application running successfully w/ nicely formatted Tables.

I included a transcript of my application in the test results directory of the release.

You don't have to mimic my output but your output should be recognizable as a list of standings.

Name all of the files in the repository with descriptive names.

Be sure your code is organized indented appropriately, uses meaningful variable names, and includes comments that aid in understanding your code.

Readability of your code will weigh into the score you receive on the assignment.