

Programming Assignment IV (*This is the last assignment*)

DUE: Saturday, May 10, 2014, 23:00. [Hard deadline. Ninova HW submission closes automatically. Late homeworks will NOT be accepted.]

NOTE: If you have any questions about the homework, contact the research assistant, Çağatay Koç, by email (kocca@itu.edu.tr) or in person (in Research Lab 3, EEB 4313).

NINOVA SUBMISSION: You should be aware that the Ninova system clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructor with your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your code, you should do so before the Ninova submission system closes. Your changes **will not be accepted by e-mail**. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. **You should not risk leaving your submission to the last few minutes.**

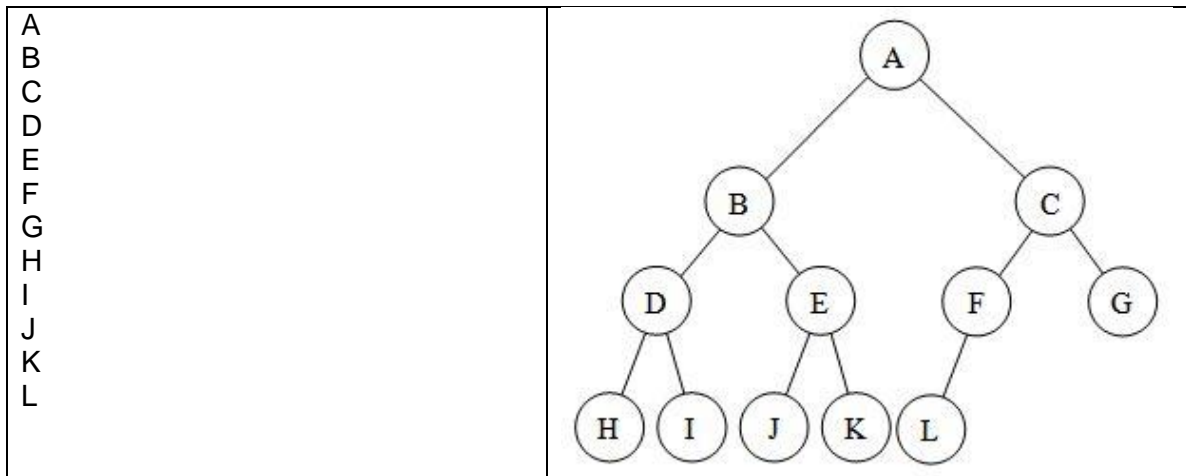
CHEATING: This is not a group assignment. It should be done individually. When a student receives information from another person about a program, it is considered cheating when the information is enough to precisely describe the code in a nontrivial part of the program. The most common example of cheating occurs when a student copies all or part of a program from another student and then changes the names of some of the program variables and functions. If cheating is discovered, a report will be made recommending a course grade of "VF".

Assignment Description

In this assignment, you will implement an application that reads binary tree data from a file and writes to an output file using the given tree traversal algorithm (inorder, preorder, or postorder). You should successfully **deallocate** all of the allocated memory before termination of your program.

- Your program should run from the command line with an input file name, an output file name, and the traversal algorithm that will be used while writing the data to the output file.
- Example commands:
 - ./yourApplication input1.txt output1.txt inorder
 - ./yourApplication input2.txt output2.txt preorder
 - ./yourApplication input3.txt output3.txt postorder
- Input files are created by breadth-first traversal of the tree. Your program should read the data and place the nodes in appropriate locations in the tree. You can assume that the given tree is **complete**.
- A binary tree T with n levels is **complete** if all levels except the last are completely full and the only missing nodes at the last level are on the right.

- An example input file and a complete binary tree of the input:



SUBMISSION PROCEDURE:

1. The top lines of all the files in your project should be comments with the following information

```
// Your Lastname, Your Firstname
// Your Student ID
// Programming Assignment 4
```
2. Write down the appropriate compiling and running codes to a readme.txt file (only if necessary).
3. Make sure that **GNU C++ compiler (g++)** compiles your project, and the application runs in Linux smoothly. This is important because we will evaluate your homework in Linux using g++. You can use the **ITU ssh server** to compile and test your application.
4. After making sure that everything compiles smoothly, archive all files into a zip file.
5. Upload this file by **23:00 on May 10, 2014** to Ninova. No late assignments will be accepted.

Your program will be run and graded on programming style and how it performs the required computation. With regard to programming style, we expect the following: (i) **comments and appropriate variable names** should be used to make your program more readable; (ii) appropriate prompts and messages for input and output should be given to the user of your program. In general, more emphasis will be placed on program clarity than on program speed or size.