Assignment 3
Deadline: 07 December 2021 (11:59 PM)

Overview

Sky Electric Company, Islamabad has hired you as a software engineer. The company has asked you to develop a software solution to manage the log files generated by their systems. The log files contain the technical details of the issues encountered during operation by the respective systems.

You have gathered the following information after interviewing the support staff of the company:

1. There are different categories of log files, e.g., Voltage Log, Power Outage Log, etc.
2. The company wants to be able to:
    a. View the file directory tree in different ways (Pre Order, In Order, Post Order, Level Order, and Tree View).
    b. Search a specific log file using quick search interface (described below).
    c. Merge two directories.
    d. Add a log file/directory.
    e. Delete a log file/directory.
    f. Calculate size of a given log file/directory.

You have decided to develop a "File System Shell" to keep track of the log files and provide the requested features.

The File System Shell

Design

You are required to generate an n-ary tree from the given directory structure, where "logs" directory becomes the root of the tree (see Appendix I) such that:

- Sub directories and files become children and descendants of the root, structured exactly as in the "logs" directory.
- Files should always be the leaf nodes
- Any number of new files and directories can be added to the tree
- Existing files and directories can be deleted from the tree
- Existing directories can be merged.

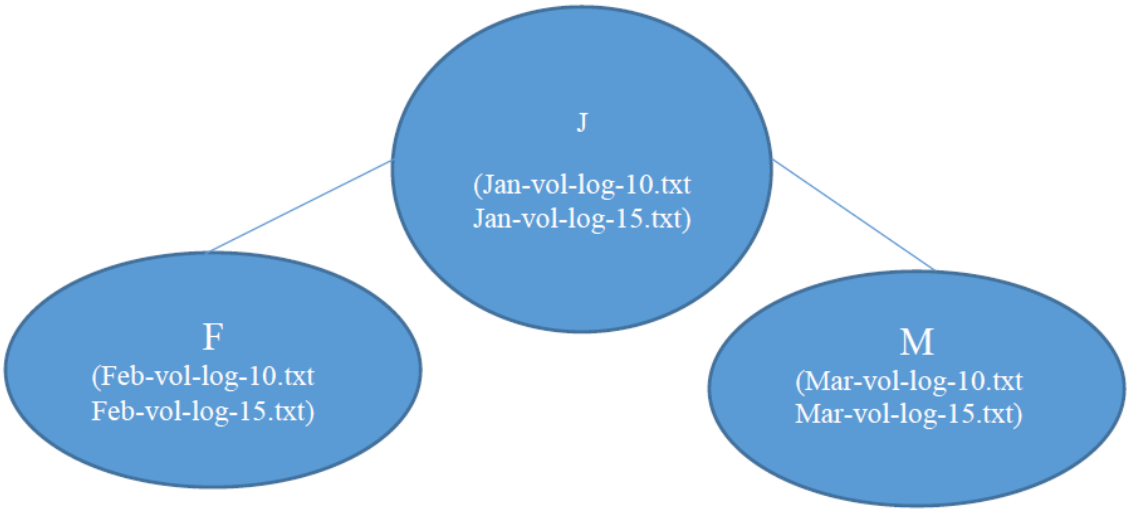
Reporting

To manage performance of the system, Sky Electric Company wants to view the log files in pre-order, post-order, in-order, level-order, as well as in the tree form. An example the tree form output is shown below:

```
logs
├── power outage logs
│   └── old-logs.txt
│   └── June
│       └── June-power-log-10.txt
│       └── June-power-log-12.txt
└── voltage logs
    └── January
        ├── Jan-vol-log-10.txt
        └── Jan-vol-log-15.txt
    └── February
        ├── Feb-vol-log-10.txt
        └── Feb-vol-log-15.txt
    └── March
        ├── Mar-vol-log-10.txt
        └── Mar-vol-log-15.txt
```

Quick Search Tree

For quick search you are required to construct (and keep updated) an AVL Tree to provide a quick search interface. AVL of the above example is given below:



In the above AVL, it is easy to navigate to a specific file using alphabetical search. It is very similar to finding a key term using index of a book. The first character of the search string is compared with the character in root of AVL, and based on the value, it is decided to get path of the file from the list of files in the current node or go to left or right. AVL returns the path in n-ary tree of the required file. AVL tree implementation must be based on templates.

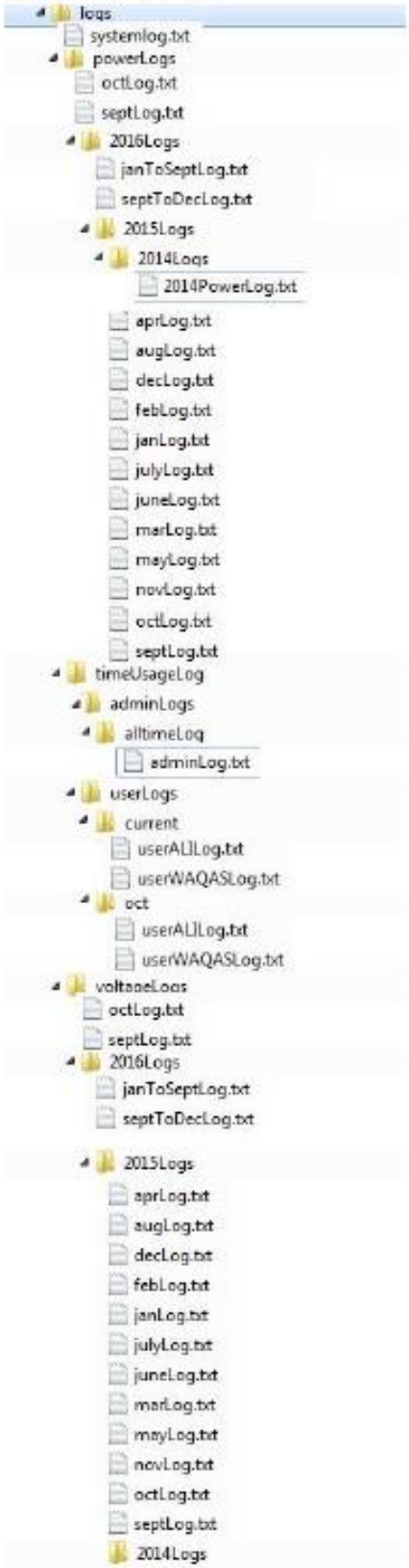
Operations

- **Display Tree:** Already Discussed in The Directory Tree above.
- **Delete File/Folder:** This should update the quick search tree and actual directory accordingly. Note: File names in different directories can be same.
- **Merge Folders:** Merging copies all the contents of one directory into another. This should update quick search tree and directory tree.
- **Display Size:** This function shall calculate and show the size of any file or a directory. You are not allowed to store size of a directory, i.e., only files know their size.
- **Search File:** Already Discussed in Quick Search Tree.

Submission Criteria & Guidelines

1. Submission: You are required to use Visual Studio 19 or above for the assignment. Combine all your work in one .h file named ROLL_NUM_A_02.h (e.g., 20i-1234_A_02.zip). DO NOT SUBMIT COMPLETE PROJECT. Move .h file to folder as ROLL_NUM_A_02 then .zip file. Submit the .zip file in the classroom within a given deadline. Failure to submit according to the above format would result in ZERO marks.
2. Path of files must be the same as the Project path. DO NOT USE ABSOLUTE PATH.
3. Code must be generic. Try to use template.
4. Use pointer-based implementation only. Array is not allowed.
5. For convenience, using string data type is allowed, but do not use STLs or any built-in functions.
6. If the required output is generated, you will be awarded full marks. Failing to generate the correct output will result in zero marks (black box checking only).
7. For syntax errors, no marks will be awarded.
8. Plagiarism cases will be dealt with strictly. If found plagiarized, both involved parties will be awarded zero marks in this assignment. Copying from the internet is the easiest way to get caught!
9. Deadline: Correct and timely submission of assignment is the responsibility of every student. Therefore, no relaxation will be given to anyone.
10. If a test case is found out to have errors, report to get test cases resolved quickly and updated in the submissions tab with a notice.

Appendix I


logs
systemlog.txt
powerLogs
octLog.txt
septLog.txt
2016Logs
janToSeptLog.txt
septToDecLog.txt
2015Logs
2014Logs
2014PowerLog.txt
aprLog.txt
augLog.txt
decLog.txt
febLog.txt
janLog.txt
julyLog.txt
juneLog.txt
marLog.txt
mayLog.txt
novLog.txt
octLog.txt
septLog.txt
timeUsageLog
adminLogs
alltimeLog
adminLog.txt
userLogs
current
userALlLog.txt
userWAQASLog.txt
oct
userALlLog.txt
userWAQASLog.txt
voltageLogs
octLog.txt
septLog.txt
2016Logs
janToSeptLog.txt
septToDecLog.txt
2015Logs
aprLog.txt
augLog.txt
decLog.txt
febLog.txt
janLog.txt
julyLog.txt
juneLog.txt
marLog.txt
mayLog.txt
novLog.txt
octLog.txt
septLog.txt
2014Logs