# In Memory File System

## Problem

Implement the basic *nix file system operations, such as ls, mkdir -p, add file, etc., of an in-memory file system.
An in-memory filesystem only exist in memory and is not persisted on disk.

## Instructions

### Building the binary

For Linux and Mac users, you can use the included Makefile to compile and run the project.

```
# Compile only
make
# Compile and run
make run
```

For Windows Visual Studio users, a vcproj template is included. Alternatively you can compile and run with the included Makefile.vs in Developer Command Prompt:

```
# cd to project directory
# Compile only
nmake -f Makefile.vs
# Compile and run
nmake -f Makefile.vs run
```

Please implement `Ls`, `MkdirP`, `AddFileWithContent`, and `GetFileContent` in `FileSystem` class. For the sake of simplicity, you can assume all file or directory paths are absolute paths which begin with `/` and do not end with `/`. You can also assume that all operations will be passed valid parameters and users will not attempt to retrieve file content or list a directory that does not exist.

Please do not implement using local filesystem, such as the `DIR`, `fstream` and `iostream`.

For example,

```
# assumption: all path starts with / and not ending with /
FileSystem fs = FileSystem();
```

```
// should print []
cout << fs.Ls("/") << endl;
fs.MkdirP("/a/b/c");
fs.AddFileWithContent("/a/b/c/d", "hello world");
// should print [a]
cout << fs.Ls("/") << endl;
// should print [d]
cout << fs.Ls("/a/b/c") << endl;
// should print [d]
cout << fs.Ls("/a/b/c/d") << endl;
// should print hello world
cout << fs.GetFileContent("/a/b/c/d") << endl;
```

## Submission

Upon completion, please follow the instructions described in the website (where you found the instructions to download the project) to submit your solution. You can submit as many times as you prefer. Your last submission will be used for evaluation as well as marking the end of your coding assessment.

Lastly, do not be concerned if you are running a little bit over time (0-10 minutes). We do not penalize moderately tardy submissions.