**Final Exercise – day 1:**

* Implement an in memory file cache .

It will be implemented as a module and will expose the following operaiotns :

* + list(path) - Return an array of all the files and directories under the given path in all levels.
  + getFile(path, filename) - Returns the content of the requested file from a memory container, and if it does not exist in memory upload it from disk , store in memory and return it.
  + saveFile(path, filename, content) - saves or update the file content
  + deleteFile(path)
  + reset – clear the memory of the modules

To make the API terse, it has to comply with the following rules:

* + ‘saveFile’ will create the full path if the destination folder doesn’t exists.
  + Only a single copy of each content of different will be kept in memory (same content different files).
  + Delete a leaf folder if it contains no files when ‘deleteFile’ is called.

For simplicity assume:

* All variables are strings.
* No need to validate input types.

You have the following repository with tests. All implementations are blank.

<https://github.com/tamarstern/InMemoryFileOPerations>

1. Your first task is to implemented the module, with only asynchronous operations , and make the tests pass.
2. Your second task is to re-write the tests to use generators.