

## Description

We know you know what a file manager is. Remember Windows Explorer? You can view files and folders in a directory and navigate between folders using icons or by entering a directory path.

Our file manager will do the same things, but with one exception: it'll be a text-based manager, not a visual one. You'll control the manager with text commands instead of clicking on icons and interacting with menus — *so old school*.

Let's start with implementing the navigation feature. It enables us to go from one directory to another. In the later stages, we can perform some actions on files within these directories.

Your file manager should be able to execute the following commands:

- `pwd` tells you in which directory you currently are
- `cd` changes the current directory

Moreover, the `cd` command can have the following attributes to specify in which directory the file manager should move to:

- `cd C:\folder1\folder2` moves from the current directory to the directory with the absolute path of `C:\folder1\folder2`. In other words, after this command is entered, the current directory should be changed to `folder2` located on disk `C:` inside `folder1`;
- `cd ..` moves one directory up. That is if the current directory is `C:\folder1\folder2`, the current directory is changed to `C:\folder1`;
- `cd folder2` (when the user is in `C:\folder1`) moves to the relative file path. The current directory should be changed to `C:\folder1\folder2`.

## Objectives

In this stage, your program should:

1. Print the prompt message: `Input the command`
2. Take the command from the user.
  - If the command is `pwd`, print the absolute path of the current directory.

Note: The starting directory is the one where your file manager program is

located. For example, if your file manager program has an absolute path `C:\Users\user\Projects\FileManager\manager.py`, and the first user command is `pwd`, the file manager should print `C:\Users\user\Projects\FileManager`.

- If the command is `cd ..`, move one level up from your current directory. Print the name of the new current folder.
- If the command is `cd path\path\path` (that is, `cd` with a relative folder path), move to the folder specified by the user. Print the name of the new current folder.
- If the user inputs any other command (including `cd` with no arguments), print the error message: `Invalid command`.
- In case the specified file or directory does not exist, handle the error using `FileNotFoundError` and print an error message of your choice. After the error message, the program should ask for input again.

3. Keep on taking new commands from the user until they type `quit`. If the user types `quit`, quit the program.

## Example

The greater-than symbol followed by a space (`>` ) represents the user input. Note that it's not part of the input.

### Example 1:

```
Input the command
> pwd
C:\Users\User\PycharmProjects\File Manager
> cd ..
PycharmProjects
> pwd
C:\Users\User\PycharmProjects
> cd ..
User
> pwd
C:\Users\User
> cd C:\Users\Documents
Documents
> cd C:\Users\Documents\Report
Report
> cd ..
Documents
```

```
> cd
Invalid command
> ppp
Invalid command
> pwd
C:\Users\Documents
> quit
```