Homework 6

Reading, writing and organizing files

Given February 26th – Due March 10th

Exercise 1. Multiclipboard

Say you have the boring task of filling out many forms in a web page or software with several text fields. The clipboard saves you from typing the same text over and over again. But only one thing can be on the clipboard at a time. If you have several different pieces of text that you need to copy and paste, you have to keep highlighting and copying the same few things over and over again.

Write a Python program to keep track of multiple pieces of text. The program will save each piece of clipboard text under a keyword. For simplicity you can save this file as mcb.py (mcb of multiclipboard).

For example, when you run:

```
python3 mcb.py save spam
```

the current contents of the clipboard will be saved with the keyword spam. This text can later be loaded to the clipboard again by running:

```
python3 mcb.py spam
```

If the user forgets what keywords they have, they can run:

```
python3 mcb.py list
```

to see a list printed in the terminal of all keywords available.

Here's what the program does:

- The command line argument for the keyword is checked.
- If the argument is save, then the clipboard contents are saved to the keyword.
- If the argument is list, then all the keywords are printed.
- Otherwise, the text for the keyword is copied to the clipboard.

Then add a delete and deleteall commands to delete one keyword or all of them.

Exercise 2. Renaming Files with American-Style Dates to European-Style Dates
Say your boss emails you thousands of files with American-style dates (MM-DD-YYYY) in their names and needs them renamed to European-style dates (DD-MM-YYYY). This boring task could take all day to do by hand! Let's write a program to do it instead.

Here's what the program should do:

- Search all the filenames in the current working directory for American-style dates.
- When one is found, it renames the file with the month and day swapped to make it European-style.

This means the code will need to do the following:

- Create a regex that can identify the text pattern of American-style dates.
- Call os.listdir() or os.walk(path) to find all the files in the working directory.
- Loop over each filename, using the regex to check whether it has a date.
- If it has a date, rename the file with shutil.move().

For this project, save your code as renameDates.py.

Exercise 3. Selective Copy
Write a program that walks through a folder tree (os.walk()) and searches for files with a certain file extension (such as .pdf or .jpg). Copy these files from whatever location they are in to a new folder.

For this project, save your code as selectiveCopy.py.

Exercise 4. Deleting Unneeded Files (Extra points)
It's not uncommon for a few unneeded but humongous files or folders to take up the bulk of the space on your hard drive. If you're trying to free up room on your computer, you'll get the most bang for your buck by deleting the most massive of the unwanted files. But first you have to find them.

Write a program that walks through a folder tree and searches for exceptionally large files or folders — say, ones that have a file size of more than

100MB. (Remember, to get a file's size, you can use os.path.getsize() from the os module.) Instead of deleting them, print these files with their absolute path to the screen.

If the exercise does not say which name to save the file, save your code files as *hm6_name_surname_ex_num.py*, where *num* is the exercise number 1, 2, etc. Comment everything so we know you wrote the code! On top of your files write this multiline comment with your information:

"""

Homework 6, Exercise 1 (or 2...)
Name
Date
Description of your program.

"""