Homework 6
Decorators and Files
Out 11/9 – Due 11/24

Exercise 1. Slow down decorator (30pts).
Write a @slowDown decorator that will sleep for a certain number of seconds before it calls the decorated function. Use an optional rate argument that controls how long it sleeps. Use a default value of 1 second for the sleep duration.

Note that you can modify the @slowDown example provided in class by letting the decorator accept an optional argument that is the number of seconds to sleep, and then use the same recursive countdown(fromNumber) as the function to decorate and test the decorator. Save your code as hw6_firstname_lastname_ex_1.py.

Exercise 2. Fibonacci sequence (40pts).
The Fibonacci Sequence is the series of numbers like this:
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

The next number is found by adding up the two numbers before it.

- 2 is found by adding the two numbers before it (1+1)
- 3 is found by adding the two numbers before it (1+2),
- 5 is (2+3),
- ……

An implementation could be like this:

```
def fibonacci(num):
  if num < 2:
    return num
  return fibonacci(num - 1) + fibonacci(num - 2)
```

But the runtime performance is terrible. This is because the code keeps recalculating Fibonacci numbers that are already known.

1. Implement a @cache decorator that will save the calculations in a function attribute dictionary. Make the decorator work for functions with more than one argument.
2. Apply the @countCalls decorator introduced in class to the fibonacci function. Do you see a difference between using @cache and not using it (hint: use nested decorator)? Write your conclusion at the top of your code, in the multiline comment under "Description of your program".
3. Save your code as hw6_firstname_lastname_ex_2.py

Exercise 3. Multiclipboard (30pts).
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 copied 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 stands for multiclipboard; the shorter name makes it easy if you run it from the command line).

For example, when you run:

python3 mcb.py save spam

Then the current contents of the clipboard will be saved under 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 full list printed in the terminal of all keywords available.

Here's what the mcb program does:

1. The command line argument is checked.
2. If the argument is save, then the current clipboard contents are saved under the keyword.
3. If the argument is list, then all the saved keywords are printed.
4. Otherwise, if only a keyword is provided and no save or list is in the arguments, the text saved under this keyword is copied to the clipboard.

Submit your code files in **a zipped archive named hw6_firstname_lastname.zip**. 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.
"""
```