Homework 3

Dictionaries and structuring data, and Manipulating strings

Given February 5th – Due February 17th

Exercise 1. Number of occurrences

Develop a program that counts the number of occurrences of each letter in a string. You can use the following string to test:

- *It was a bright cold day in April, and the clocks were striking thirteen.*

You may want to use the module pprint for pretty printing of dictionaries.

```
import pprint
pprint.pprint(dictionary)
spam = pprint.pformat(dictionary) # Pretty text as a string value
print(spam)
```

Exercise 2. Tic-Tac-Toe

In this exercise we are going to create a Tic-Tac-Toe game.

1. Create the data structure
   - Nine slots that can each contain an X, an O, or a blank.
   - To represent the board with a dictionary, you can assign each slot a string-value key.
   - String values in the key-value pair to represent what's in each slot on the board:
     - 'X'
     - 'O'
     - ' '
2. Create a function to print the board dictionary onto the screen
3. Add the code that allows the players to enter their moves

(**Note**: This isn't a complete tic-tac-toe game — for instance, it doesn't ever check whether a player has won — but it's enough to see how data structures can be used in programs.)

Exercise 3. Password locker.

1. You probably have accounts on many different websites. It's a bad habit to use the same password for each of them because if any of those sites has a security breach, the hackers will learn the password to all of your other accounts.

Develop a simple password manager software on your computer where you write the account name (blog, facebook, instagram, etc) as an argument and the password is copied to the clipboard. Then you can paste it into the website's Password field.

For example, if you execute this command line in your terminal:

```
python3 ./pw.py instagram
```

The password for Instagram (example: htS:D`t*hQH3]9"C) should be copied to the clipboard.

2. Add a second argument where is a master password. Only copy to the clipboard the password for the first argument if the master password is correct. (Do not save your password manager in your program!!)

(**Note**: (You don't need to code this) Now your program is more secure than before, but it is still **insecure** because anyone could go to your source file and get the passwords. To be secure:

- You should introduce your passwords through the command line arguments.
- Once a password is introduced, encrypt them using a master password you only know (not in the source file, passed through the command line arguments) and store them in a file.
- When you want to retrieve them, decrypt them using the master password.

Then, it would be **secure** because no one can retrieve the passwords without the master password and the passwords are saved encrypted in the computer.)

Exercise 4. Fantasy Game Inventory
Imagine you have this inventory in a fantasy game:

Item	Number of this items
Rope	1
Torch	6
Gold coin	42
Dagger	1
Arrow	12

1. Save the inventory in a data structure.
2. Write a function that will take any possible inventory and display it in a pretty format.
3. Write a function to add new items to the data structure. Make sure new group of items can be created.
4. Write a function to delete items (not a whole group) from the data structure.

Submit your code files as *hm3_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 3, Exercise 1 (or 2...)
Name
Date
Description of your program.
"""