Homework 4
OOP and Regex
Out 10/7 – Due 10/23 (Midterm 10/21)

Exercise 1. Password Locker (20 pts).
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, hackers will learn the password to all of your other accounts.

Develop a simple password manager software where

1. Your account names and the corresponding passwords are stored in a data structure in the password manager;[1]
2. You provide the account name (e.g., blog, facebook, instagram, etc) as an argument, and the password is copied to the clipboard (so you can paste it into the website's password field);
3. Your code should check if the input argument is provided, otherwise it should print an error message and exit;
4. Your code should also check if the provided argument exits as an account name in the password manager, otherwise it should print an error message and exit;
5. Save your code as hw4_firstname_lastname_ex_1.py.

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

python3 hw4_firstname_lastname_ex_1.py instagram

The password for Instagram (for example: htS:D`t*hQH3]9"C) should be copied to the clipboard (don't use your real password!).

Exercise 2. Rectangle, Circle and Square (30 pts).
Write three Python classes named **Rectangle** constructed by a length and width, a **Circle** constructed by a radius, and a **Square** constructed by a side length. All classes should have the methods that compute:

- The area
- The diagonal
- The perimeter

Use as much inheritance as you can. Import the math module so you can call its sqrt() method. At the end of the code, use these classes to calculate the perimeter of a circle with a radius that's the half of the diagonal of a rectangle with a length of 20 and width 10. Save your code as hw4_firstname_lastname_ex_2.py.

[1] Note that this is VERY insecure because anyone could go to your source file and get the passwords. So please don't use your real passwords in this exercise. To be secure, the passwords should be stored encrypted, and decrypt them when they are needed (no need to implement this).

Exercise 3. Phone Number and Email Address Extractor (30 pts).

You have a boring task of finding every phone number and email address in a long web page or document. If you manually scroll through the page, you might end up searching for a long time. But if you had a program that could search the text in your clipboard for phone numbers and email addresses, you could simply do ctrl-A to select all the text, ctrl-C to copy it to the clipboard, and then run your program. It could replace the text on the clipboard with just the phone numbers and email addresses it finds.

So, your phone and email address extractor will need to do the following:
- Get the text from the clipboard.
- Find all phone numbers and email addresses in the text.
- Paste them to the clipboard.

Save your code as hw4_firstname_lastname_ex_3.py. Note that finding a web page and copying it to the clipboard is not part of your code.

Exercise 4. Strong Password Check (20 pts).
Write a program that uses regular expressions to make sure the password string it receives through an input argument is strong. A strong password in this exercise is defined as one that (1) is at least eight characters long, (2) contains both uppercase and lowercase characters, and (3) has at least one digit.

You may need to test the password string against multiple regex patterns to validate its strength as described above (note that please only use regexes). Save your code as hw4_firstname_lastname_ex_4.py

Submit your code files in **a zipped archive named hw4_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 4, Exercise 1 (or 2…)
Name
Date
Description of your program.
”””