

## Lab 2

Due **Wednesday 2/6/19** at 11:59 p.m. on Zybook and Canvas

You are required to work in a pair.

Lab guidelines: Complete the problems below. Submit the programming problems to Zybook for grading, and the text response to Canvas. Each program should be placed in its own file, named exactly as specified.

Your submission must be your work, and only your work. All program files should contain header docstrings with name(s) of program author(s), creation date, and a brief explanation of the program, as well as in-line comments explaining what your code does.

### 0. Generating secure password that you can remember, `lab2pr0.py`

It's not easy to come up with a secure password that one can actually remember. One of the methods proposed is to take a line of text and take the first letter of each word to form a password. An extra step is to substitute "o" with 0, "a" with "@", and "l" with 1. For instance, using "Cookies and Code, let's meet on Tuesdays" would yield "C@C,1m0T". Write a program that asks the user to enter a phrase and the program outputs the password that is generated using these rules.

*You can assume that the phrase starts with a letter, and that words in the phrase are separated by a single space or a common punctuation sign such as ".,?!;" followed by a space.*

### 1. Password check, `lab2pr1.py`

Many websites these days require that a password is between 8 and 20 characters, it does not contain spaces, and it satisfies the following conditions: It must contain at least one uppercase letter, at least one lowercase letter, at least one number, and at least one special symbol character ("!?,,:\$#\_&"). Write a program that keeps asking the user to enter the password until they enter one that satisfies the requirements.

```
Please enter a password:  
@Tulane2020  
Please enter a password:  
#Tulane2020  
Password accepted
```

### 2. Privacy program, `lab2pr2.py`

Write an automated password-hiding program that could be used in an automatic password manager. The program first reads the user passwords, one at a time, and stores them in a list. When the user enters an empty string (just hits enter), they are finished with the input. The program should then replace each password with a string of \* corresponding to the length of the original password. As the output, the program should print the original list and the resulting list.

Here is the output from a sample run of your program:

Please enter a password (press [enter] to finish): Tulane123

Please enter a password (press [enter] to finish): Bruff

Please enter a password (press [enter] to finish): LBCLBC

Please enter a password (press [enter] to finish):

```
['Tulane123', 'Bruff', 'LBCLBC']
```

```
['*****', '*****', '*****']
```

3. Codebreakers, lab3pr3.txt Watch the talk by security analyst, cyber punk, hacker pride, geek girl Keren Elazari "Hackers: the Internet's immune system" (16 minutes):

[https://www.ted.com/talks/keren\\_elazari\\_hackers\\_the\\_internet\\_s\\_immune\\_system](https://www.ted.com/talks/keren_elazari_hackers_the_internet_s_immune_system)

- (a) What is your opinion on the speaker's main argument? Do you agree that hackers are the Internet's immune system? Why or why not? Explain your point in 3-6 sentences.
- (b) There are several examples in the talk of the situations in which hackers may have done "good", but their actions were illegal according to the current law. How do you think the law should treat hackers, breakers, cyberattackers? Explain your point in 3-6 sentences.
- (c) In 3-6 sentences, summarize the possible positive and negative effects of hacker's actions. Feel free to use recent events such as the Facebook data leak, the Yahoo data breach, the Marriott data breach, or the Equifax leak for example (search for these news on Google if necessary).