

1. zyBooks Labs

Please follow the link on Canvas to complete the following zyBooks labs:

- 3.15 LAB: Input and formatted output: House real estate summary
- 3.17 LAB: Warm up: Creating passwords
- 3.18 LAB*: Program: Painting a wall
- 4.12 LAB: Smallest number
- 4.13 LAB: Interstate highway numbers

2. Simple Algorithmic Encryption

This exercise will introduce the user to simple encryption. While there are easier ways to perform this encryption, this activity is structured to give the student practice using a variety of techniques on strings and lists. Now, suppose you wanted to send a secret message to a friend. The two of you would then decide on an algorithm to encrypt the message so that the end result would not be easily read by others. The goal of this lab component is to encrypt a message in the following way:

- Append a random lowercase letter (a – z) to the end of the string.
- Move the first two characters of the string to the end of the string (which requires deleting the first characters from the start of the string).

Recall that `input()` reads a string of text from the user input (i.e., the keyboard), but strings are immutable, which means that they are read-only in nature and cannot be modified. So, we will perform our encryption lists, which are mutable!

Write a small, but complete Python3 program called **Lab4A.py** that encrypts a string entered by the user as follows:

- a. Prompt the user for and read in a string (>1 character) to be encrypted. You may assume the user enters more than one character.
- b. Since strings are immutable, we want to convert the string entered by the user to a list (which is mutable). To do so, we create an empty list by assigning `[]` to a list variable. We can then convert the string to a character-wise list using `list1[:0] = str1`, where `list1` is our list and `str1` is our string. You can print `list1` to confirm that the operation was successful.
- c. Now, let's start the encryption portion! Since we want to append a random lowercase character to the newly created list, we need to generate a random number between 97 and 122, inclusively, which is the ASCII value for lowercase characters between a through z. To do this, we use the `randint()` function in the `random` module. *Hint: There is an example of this in Dr. Thompson's public directory!* Next, use the `append()` function to add this lowercase character to the end of your list object. Note that since `randint()` generates an integer, we

must first convert this to a character using `chr()` so that it appends a lowercase character and not an integer.

- d. Next, we want to append the first two characters to the end of the list. To do this, we create a second list consisting of the first two characters of the first list by `list2 = list1[:2]`, where `list2` is our second list. Once this is done, use the `+` operator to append the second list to the first list.
- e. Since we basically copied the first two characters and appended them to the end of the list, we now need to remove the first two characters from the start of the list. We can do this using the `remove()` function, passing in the first character of the string (you can slice a string using `[0]` to access the first character). Then, repeat the `remove()` call with the second character in the string.
- f. Almost done! First, print out the now modified list to see that our output is a list and not a string. But since we want the output to be a string, we need to now convert our list to a string. To do this, we first assign the empty string `""` to a new string variable (be sure there are no spaces between the double quotes) and then use the `join()` function on the string such as `str2 = str2.join(list1)`, where `str2` is our newly created string. Finally, print the encrypted string!

For example, the output might look like this (input shown in **bold**):

```
$ python3 Lab4A.py
Enter a string (>1 character) to be encrypted: Hello
Original list: ['H', 'e', 'l', 'l', 'o']
Encrypted list: ['l', 'l', 'o', 'f', 'H', 'e']
Encrypted string: llofHe
```

Note that you will submit this file to Canvas.

3. Working with Dictionaries

This exercise will give you some basic practice working with dictionaries and branches. As a produce manager at a local grocery store, you have implemented a simple inventory system for the fruit carried at your store.

Write a small, but complete Python3 program called **Lab4B.py** that encrypts a string entered by the user as follows:

- a. Create an inventory dictionary by with three initial different types of fruit along with their quantities. For example, you might add 105 'apples', 56 'bananas', and 0 'oranges'. Note that the name of the fruit is a string, while the quantity of fruit is an integer.
- b. Print all entries in the current inventory (include both fruit name and quantity).
- c. Suppose you receive a new shipment of some fruit (e.g., 217 'pears'). Prompt for and read in the name of the fruit and the quantity. Before adding to the inventory, you must first check if the fruit already exists in the dictionary. If the fruit already exists in the dictionary, you will simply update (Hint: You can use the `+=` operator!)

the quantity of fruit in the inventory and then print out the current quantity of that fruit. Otherwise, you will add the new fruit entry to the dictionary.

- d. Print the current list of fruit and their quantities using the `items()` function.
- e. Your store has decided to no longer carry a certain type of fruit. Prompt for and read in the name of the fruit being discontinued at the store. Before deleting from the inventory, you must first check if the fruit exists in the dictionary. If the fruit exists in the dictionary, you will delete the specified entry and print that the fruit is no longer carried in the store. Otherwise, you will print a message that the fruit is already not carried in the store.
- f. Finally, print the total types of fruit carried in the store using the `len()` function.

For example, the output might look like this (input shown in **bold**):

```
$ python3 Lab4B.py
Current fruit inventory: {'apples': 105, 'bananas': 56,
'oranges': 0}
Enter latest fruit shipment received: pears
Enter quantity of pears: 217
dict_items([('apples', 105), ('bananas', 56), ('oranges', 0),
('pears', 217)])
Enter fruit being discontinued at store: oranges
oranges no longer carried in store
Total types of fruit in store = 3
$ python3 Lab4B.py
Current fruit inventory: {'apples': 105, 'bananas': 56,
'oranges': 0}
Enter latest fruit shipment received: oranges
Enter quantity of oranges: 200
Amount of oranges in store = 200
dict_items([('apples', 105), ('bananas', 56), ('oranges', 200)])
Enter fruit being discontinued at store: kiwis
kiwis already not carried in store
Total types of fruit in store = 3
```

Note that you will submit this file to Canvas.

Now that you have completed this lab, it's time to turn in your results. Once you've moved the files to your windows machine (using **WinSCP**), you may use the browser to submit them to Canvas for the **Lab 04** dropbox.

You should submit the following files:

- **Lab4A.py**
- **Lab4B.py**
- **(Note that the zyBooks labs are submitted separately through Canvas.)**

Ask your TA to check your results before submission.

Now that you've finished the lab, use any additional time to practice writing simple programs out of the textbook, lectures, or even ones you come up with on your own to gain some more experience.