CS-UY 1114 Lab 11

Key points:
- Dictionary

Part I: Pencil and paper problem:

```
def f1 (my_dict):
   temp = 0
   for key in my_dict:
       temp = temp + my_dict[key]
   return temp

def f2 (my_dict):
   temp = ''
   for key in my_dict:
       if temp < key:
           temp = key
   return temp

def f3(my_dict,k,v):
   if k in my_dict:
       my_dict[k]=v

def main():
   a_dict={'bill':1,'rich':2,'fred':10,'walter':20}

   print(f1(a_dict)) # Line 1
   print(f2(a_dict)) # Line 2
   print(None == f3(a_dict,'bill',-1)) # Line 3
   print(a_dict) # Line 4

main()
```

(a) What output is produced by Line 1 of the program?
(b) What output is produced by Line 2 of the program?
(c) What output is produced by Line 3 of the program?
(d) What output is produced by Line 4 of the program?

Part II: Programming problem:
May I Take Your Order?

a) Write a function, `read_menu(filename)` that is given a name of a file that contains a menu of a restaurant. This function should read that file, and create and return a dictionary that represents the menu in the file. The keys of the dictionary are the name of the food, and the values are their corresponding prices.

Note: Assume that the food name and the price, in the input file, are separated by a ':' symbol and a space.

For example, for a file that looks like:

Big Mac: $3.99

Big Mac - Meal: $5.99

2 Cheeseburgers: $2.00

2 Cheeseburgers - Meal: $4.89

Quarter Pounder with Cheese: $3.79

Quarter Pounder with Cheese - Meal: $5.79

.

.

.

A call to `read_menu` should return a dictionary that could look like:

{'Big Mac': 3.99, '2 Cheeseburgers': 2.0, 'Quarter Pounder with Cheese': 3.79, 'Big Mac - Meal': 5.99, 'Quarter Pounder with Cheese - Meal': 5.79, '2 Cheeseburgers - Meal': 4.99, …}

b) Write a function `read_customer_order()` that prompts the user to enter a sequence of items that he/she would like to order, followed by the word "done". A call to `read_customer_order` should return a list of strings, one for each food.

For example, a call to `read_customer_order` should interact with the user as follows:

What would you like to order?

Quarter Pounder with Cheese - Meal

Big Mac

done

In this case the function should return: ['Quarter Pounder with Cheese - Meal', 'Big Mac']

c) Write a function `compute_price(menu_dictionary, order_list)` that takes a dictionary representing a menu, and a list of strings representing the customer's order and returns the sum of the prices of the items the customer ordered.

For example, given the example inputs from (a) and (b), your function should return 9.78.

d) Write a `main()` function that calls the other functions to prompt the user to enter a name of a file that contains a menu, then prompts three customers to enter their orders and prints the total price for each of them along with 8.5% tax.

Part III: Pair-programming problem:

In this exercise, implement simple phonebook operations. Phonebook is a dictionary with names as keys and phone numbers as values. A valid phone number is a string with 10 digits. For example, '2014567890' is a valid phone number, but '201a45b789' or '20145678' are both invalid.

Your program should have the following functions:

a. A function `add_entry(phonebook, name, phonenumber)` that adds an entry to a `phonebook`. This function should take a dictionary `phonebook`, a string `name` and a string `phone_number` as arguments. It then adds an entry with key `name` and value `phone_number` to `phonebook` given that neither of the following is true:

- There exists an entry in `phonebook` with key `name`.
- `phone_number` is not valid.

Your function should display an error message if the entry cannot be added, and states the reason.

Hint: you might want to write a helper function to check if `phone_number` is valid.

B. A function `lookup(phonebook, name)`looks up `phonebook`. This function should return the phone number associated to `name` in `phonebook`. If the input `name` is not found in `phonebook`, display an error message that indicates so.

c. A function `print_all(phonebook)` that prints all the entries in `phonebook`. Your function should print both the name and the phone number.

d. A main function that opens ‘Lab12-phonebook.txt’ in which each line has name and phone number. The names are in the form ‘Last, First’ (Note: some of the phone-numbers could be bad format and some of the names could be duplicates.) Use the your function in a to construct a phonebook from entries in ‘Lab12-phonebook.txt’.