Phone Contacts

1 Overview

You and your partner will be constructing a program that works like the contacts in your phone. You should be able to add new contacts and update information for existing contacts. To do so, you will need to create a data structure that stores the name, email, phone numbers and age for each contact. You should be able to add and update contacts using their first and last name. You will create several functions to complete this task as described in section 4.

2 Learning Outcomes

By the end of this project students should be able to:

- Read and write programs that define simple functions
- Read and write programs using lists and dictionary syntax
- Work effectively with a partner using pair-programming
- Write an effective report that describes the students' problem solving process

3 Pre-lab Instructions

Do this part before you come to lab:

- Read Problem Space Chapter 6: Collections.
- Read the full lab instructions and plan your strategy for representing your contacts. Think about the requirements and how your data organization can satisfy the requirements. Will you use a list, a dictionary, or maybe both?
- Do the following for your pre-lab and be prepared at the beginning of lab to show it to the lab aide.
    1. Think about how you would use multiple types of data structures. How would you create a dictionary that holds a list?
    2. Describe your plan (i.e. what data you need to store, how you plan to store each part of the required data). Draw an example with information for three contacts in your phone.
    3. Look up the "in" operator in Python and write a couple of sentences about how you think you would use it in this lab.
    4. Look at the function descriptions below and write signatures for each of them. You will implement the function signatures in your lab.

4 Lab Instructions

Do this part in lab:

You should implement several functions that the Phone Contacts program will need. Each function with take in data through the parameters (i.e., variables names listed in parentheses) and make updates to data structure that holds the contact information, return, or print information for a particular contact. The parameters needed are listed in the table below, and each function is described below that.

1. Make an empty dictionary and call it 'contacts'. Where in your code would you implement this? Think about how and why we write functions. What do functions help us do? Would you want to put your dictionary in the global space? Why or why not?

2. Using the function descriptions and parameters given below, implement these functions to work with your dictionary

3. Test your code using the test code given below. This code should be implemented in a main() function and will test that you have coded your functions correctly. Keep in mind, some of your functions will return a value when called, and some won't. One function will produce "output" or will print something out. Pay attention to the descriptions given. Don't forget to call main() to start your program.

Parameter definitions: # used consistently throughout the program

Parameter Name	Parameter Description
contacts	The data structure you will use to hold your contacts
first	the first name of the person, which will be part of the key used to add and/or edit a contact's information
last	the last name of the person, which will be the other part of the key used to add and/or edit a contact's information
email	the email address of the person
phone	the phone number of the person
age	the age of the person

For each of these functions, remember to add comments to your code describing in your own words what they do

create_contact function

Parameters (put the parameters in your function in this order): contacts, first, last, email, age, phone

The create contact function will take in your data structure, a key variable (first and last name) that will have an attached email, age and phone number associated with it. This function should add the contact information to your data structure of contacts.

It does not return anything.

update_contact_number function

Parameters (put the parameters in your function in this order): contacts, first, last, phone

This function will take in your data structure, a key (first and last name) and phone number, and update (change) the phone number of the contact in question.

It does not return anything.

update_contact_email function

Parameters (put the parameters in your function in this order): contacts, first, last, email

This function will take in your data structure, a key (first and last name) and email address, and update (change) the email of the contact in question.

It does not return anything.

update_contact_age function

Parameter (put the parameters in your function in this order): contacts, first, last, age

This function will take in your data structure, a key (first and last name) and age, and update (change) the age of the contact in question.

It does not return anything.

get_contact_email function

Parameters (put the parameters in your function in this order): contacts, first, last

This function will take in your data structure, a key (first and last name), and return the email address of the contact.

Returns: email address

get_contact_age function

Parameters (put the parameters in your function in this order): contacts, first, last

This function will take in your data structure, a key (first and last name), and return the age of the contact.

Returns: the age of the person

get_contact_number function

Parameters (put the parameters in your function in this order): contacts, first, last

This function will take in your data structure, a key (first and last name), and return the phone number of the contact.

Returns: the phone number of the person

contains_contact function

Parameters (put the parameters in your function in this order): contacts, first, last

This function takes in your data structure, and will search for the requested key. If the key exists, it should return True. If it does not exist, it should return False.

Returns: True if contact exist, else returns False

display function

Parameters (put the parameters in your function in this order): contacts, first, last

This function will take in your data structure and key and should print all the information about the existing contact. It should be formatted in the following manner:

Sarah Sanders
Email: sarah.sanders@nau.edu
Phone: 593-026-2532

Age: 18

Make is it so that the contact information is displayed only if the contact exists. If the contact does not exist, print a statement to the console telling the user so.

Output: Prints all data for a contact in the specified manner.

5 Test Code

Below is the test code you should use in the main() function of your program. If you find that the output prints "Failed", it means that one of your functions is not working correctly.
Be sure to call the main function to start the program. You may also write your own test code utilizing the functions above to test your program. Then complete your lab report according to the general lab report guidelines.

```
    # be sure to create the empty 'contacts' structure before executing the code below

    create_contact(contacts, "Katie", "Katz", "katie.katz@nau.edu",
25, "857-294-2758")
    create_contact(contacts,"Jim", "Jones", "jim.jones@nau.edu", 19,
"525-866-2749")
    create_contact(contacts,"Sarah", "Sanders",
"sarah.sanders@nau.edu", 18, "593-026-2532")

    print("Creation of Jim Jones: {}".format(
        "Passed" if contains_contact(contacts,"jim", "Jones")
         else "Failed"))

    print("Creation of Katie Katz: {}".format(
        "Passed" if contains_contact(contacts,"Katie", "kaTz") else
        "Failed"))

    print("Creation of Sarah Sanders: {}".format(
        "Passed" if contains_contact(contacts,"Sarah", "Sanders") else
        "Failed"))

    update_contact_age(contacts, "Sarah", "Sanders", 19)
    print("Updating Sarah Sanders age to 19: {}".format(
        "Passed" if get_contact_age(contacts,"sarah", "sanDers") == 19
         else "Failed"))

    update_contact_email(contacts, "Jim", "Jones",
"jim.jones@gmail.com")
    print("Updating Jim Jones's email: {}".format(
        "Passed" if get_contact_email(contacts,"jim", "jones") ==
        "jim.jones@gmail.com" else "Failed"))
```

```
update_contact_number(contacts,"Katie", "Katz", "907-536-2946")
print("Updating Katie Katz's number: {}".format(
    "Passed" if get_contact_number(contacts,"Katie", "Katz") ==
    "907-536-2946" else "Failed"))

 display(contacts, "Katie", "Katz")
 display(contacts, “George”, “Shaw”)
```