CSC 110 – Lab 4b
Writing Programs with Functions

Name: Joey Koumjian

Lab Objectives:

- Write a main function for a program, given the other function definitions
- Apply modular design by breaking a big problem into smaller problems
- Write Python programs using functions

Lab Policy:

- Please complete all sections of the lab, in order.
- Work with your partner using the rules of Pair Programming.
- When you get to an *Instructor Signature*, raise your hand. Do not move on until you get signed off.
- If you finish all required parts of the lab, move on to the Challenge Problems.
- If you finish all Challenge Problems, you may work on your homework that is due on Monday.
- You may not leave until the lab period is over.

Main Function

(Points 4)

1) Given the following skeleton code for a program that finds the average grade for a computer science class, write the main function here on this page.

```
def getGrades():
     return gradeList

def computeAverage(gradeList):
     return average

def printResults(average):
     return

def main():
     gradeList = getGrades()
     average = computeAverage(gradeList)
     printResults(average)
```

Instructor Initials _____

2) Suppose you were asked to compute the average grades for a computer science class and a math class. How would you modify the main function above to do this?

Use a loop to run everything inside the function twice so you can have one value for computer science and one for math. You could do this for 2 specific classes (i < 2) or use a variable like n to do n amount of classes (i < n).

```
def main():
     i = 0
     while i < 2:
          gradeList = getGrades()
          average = computeAverage(gradeList)
          printResults(average)
          i = i + 1
```

Instructor Initials _____

Sequential Search

(Points 6)

Recall the Sequential Search program that we wrote in class. You can find it on the Sample Code page of the Brightspace site (Week 3 - Lists (Thu)). We are going to convert this program into a program with functions.

1) Look at the code and think about what this program does. We need to break this program into a set of subtasks that will become our functions. What are the subtasks involved in this program? (HINT: There are three subtasks)

    a. Get data function

    b. Search function

    c. Print results function

2) For each subtask, answer the following questions (fill in the table below):

    What is the function's job?

What are the function's parameters? That is, what information does it need in order to do its job?

What will it return? That is, what information is it computing that it will return to the function that calls it?

Function Name	Job	Parameters	Return
getData	Gets the values of names and phone numbers and appends them to separate lists	none	names list and phone numbers list
search	searches through the lists to find what phone number you searched for	names list and phone numbers list, and who you search for	the name of who you searched for
printResults	prints the name of who you searched for	the name of who you searched for	nothing

Instructor Initials _____

3) For each of the functions, write the function skeleton. That is, write the definition with parameters, and the return statement. You are NOT writing the function yet. Write the skeleton code here:

```
def getData():
    return nameList, phoneList
def search(nameList,phoneList,who):
    return name
def printResults(name):
```

Instructor Initials _____

4) Write the main function. Using the function skeleton, you should be able to write the main function that calls the other functions. Note: You can assume that the main function asks the user to enter a phone number to search for.

```
def main():

    nameList, phoneList = getData():

    who = input("Enter the number to search for: ")

    name = search(nameList,phoneList,who):

    printResults(name):
```

Instructor Initials _____

5) Open a new file in IDLE and type all of the code that you have written above. Now you will need to fill in the code for each of the functions. Write them one at a time and test them separately.

6) Run the entire program and test it to make sure that it works when the number you are searching for is in the list, and when it is not in the list.

   Name your program file `lab4b_1.py` and submit it to Gradescope under the assignment named Lab 4b-1

   Your output should look like this:

```
>>> main()
How many names in the list: 5
Name: Alice
Phone: 555-1212
Name: Betty
Phone: 456-9876
Name: Clara
Phone: 789-6543
Name: Debbie
Phone: 333-5555
Name: Elaine
Phone: 998-7766
Enter phone number to search for: 789-6543
Phone number belongs to  Clara
>>> main()
How many names in the list: 5
Name: Alice
Phone: 555-1212
Name: Betty
Phone: 456-9876
Name: Clara
Phone: 789-6543
Name: Debbie
Phone: 333-5555
Name: Elaine
Phone: 998-7766
Enter phone number to search for: 444-5555
Phone number not found
```

Instructor Initials _____

CHALLENGE PROBLEMS

Every lab assignment, except for Lab 1, has at least one Challenge Problem. These exercises are meant to be a bit more challenging than the required work for the lab activity. These problems are meant to be done after you and your partner finish the required portion of the lab. By the end of the semester, every student is required to complete at least 4 Challenge Problems. This will count as a full lab grade. Any Challenge Problems completed beyond the required 4 will count as extra credit.

1) Go to projecteuler.net and look at Problem 6 – Sum Square Difference. Solve this problem. Make sure that you use at least 2 functions in your solution, and that your solution will be able to find the difference between the squares of the first n natural numbers and the square of the sum.

Instructor Initials: _____

2) Look at Problem 34 on projecteuler.net, Digit Factorials. Solve this problem. Make sure that you properly define the functions for this solution. HINT: It should not be one function. Also, for practice sake, do not use built-in python functions like sum.

Instructor Initials: _____