CS160 Computer Science I
Lab 9

Objectives
Work with lists
Work with functions
Work with files

Assignment
Write each of the following functions. The function header must be implemented **exactly** as specified. Write a main function that tests each of your functions.

Specifics
In the main function ask for a filename and use the required function to fill a list with the values from the file. Each file should have one numeric value per line. This has been done numerous times in class, and in a previous lab assignment. You can create the data file using a text editor or the example given in class – **do not** create this file in this program. Then call each of the required functions and then display the results returned from the functions. Remember that the functions themselves will not display any output, they may return values that can then be written to the screen, or they may alter the list passed to the function.

If there are built in functions in Python that will accomplish the tasks lists below YOU CANNOT USE THEM. The purpose of this lab is to get practice working with lists, not to get practice searching for methods. DO NOT sort the list, that changes the order of the items. The order of the values in the list should be the same as the order of the values in the file.

Do not alter the list unless the function specifically asks you to alter the list.

Do not return a list, or any value, unless the function specifically asks you to do so.

Your main program can ask for input, but it does not have to. It DOES need to test all of your functions, and more than once, with different data each time.

Exit the program immediately if the file name specified does not exist.

Required Functions

`def fillList (fileName)` – Creates, fills, and returns a list with values read from the specified file. Returns `None` is filename does not exist.

`def printList (message, theList)` – Prints the message on one line, all of the values on a single line, with each value taking 4 spaces total, and then a blank line. This function prints a total of 3 lines of output when called.

`def updateList (theList, change)` – Adds change to each value in `theList`. This function does not return anything.

`def mergeLists (theList1, theList2)` – Creates a new list, appends the values from `theList1` and `theList2` (in that order) and returns the new list.

`def adjustList (theList)` – Adjusts the values in `theList` such that if any value is less than 0 it is adjusted to 0, or if any value is greater than 100 it is adjusted to 100. Otherwise the value is not altered. This function does not return anything.

`def inAscendingOrder (theList)` – Returns `True` if the list is in ascending order, otherwise returns `False`.

`def inDescendingOrder (theList)` – Returns `True` if the list is in descending order, otherwise returns `False`.

`def matchingValues (theList, lowLimit, highLimit)` – Creates, fills, and returns a list with all of the values from `theList` that fall within the inclusive range of `lowLimit` and `highLimit`. DO NOT alter the original list.