

### **Programming Assignment 3**

#### **ECS 10 - Summer 2017**

**All solutions are to be written using Python 3. Make sure you provide comments including the file name, your name, and the date at the top of the file you submit. Also make sure to include appropriate docstrings for all functions.**

**The names of your functions must exactly match the names given in this assignment. The order of the parameters in your parameter list must exactly match the order given in this assignment. *All loops in your functions must be while loops. If you've been reading ahead, you may have discovered lists and conclude that a list may be the key to solving one or more of the problems below. That would be an incorrect conclusion; don't use lists in this programming assignment. While we're talking about things you should not use, do not use the Python functions `sum`, `min`, or `max`. Please note that in all problems, the goal is to give you practice using loops, not finding ways to avoid them. Every solution you write for this assignment must make appropriate use of a while loop.***

**For any given problem below, you may want to write additional functions other than those specified for your solution. That's fine with us.**

**One other thing: if you haven't already done so, or even if you have, go to our Course Information page on Canvas and read the section on Academic Misconduct. Submitting other people's work as your own, including solutions you may find on the Internet, is not permitted and will be referred to Student Judicial Affairs.**



### Problem 3

Create a Python function called `printHashtags` that expects one argument, a non-negative integer, and *prints* a row of hashtags, where the number of hashtags is given by the value passed as the argument. Control the printing with a while loop, not with a construct that looks like `print("#" * n)`. This function does not return any value. Here are some examples of how your function should behave:

```
>>> printHashtags(1)
#
>>> printHashtags(2)
##
>>> printHashtags(3)
###
>>> printHashtags(4)
####
>>> printHashtags(0)

>>>
```

### Problem 4

Create a Python function called `printTriangle` that expects one argument, a non-negative integer, and *prints* a right triangle, where both the height of the triangle and the width of the widest part of the triangle are given by the value passed as the argument. This function does not return any value. Your function should call the function from Problem 3 in printing the triangle. Here are some examples of how your function should behave:

```
>>> printTriangle(4)
####
###
##
#
>>> printTriangle(3)
###
##
#
>>> printTriangle(2)
##
#
>>> printTriangle(1)
#
>>> printTriangle(0)

>>>
```

## Problem 5

Create a function called `allButMax` that expects no arguments. Instead, this function gets its input from the user at the keyboard. The function asks the user to enter a series of numbers greater than or equal to zero, one at a time. The user types `end` to indicate that there are no more numbers. The function computes the sum of all the values entered except for the maximum value in the series. (Think of this as dropping the highest homework score from a series of homework scores.) The function then both prints the sum and returns the sum. You may assume the user inputs are valid: they will either be a number greater than or equal to zero, or the string `end`. Here are some examples of how your function should behave:

```
>>> allButMax()
Enter next number: 20
Enter next number: 30
Enter next number: 40
Enter next number: end
The sum of all values except for the maximum value is: 50.0
50.0
>>> allButMax()
Enter next number: 1.55
Enter next number: 90
Enter next number: 8.45
Enter next number: 2
Enter next number: end
The sum of all values except for the maximum value is: 12.0
12.0
>>> x = allButMax()
Enter next number: 3
Enter next number: 2
Enter next number: 1
Enter next number: end
The sum of all values except for the maximum value is: 3.0
>>> print(x)
3.0
>>> allButMax()
Enter next number: end
The sum of all values except for the maximum value is: 0
0
```

## Problem 6

Create a function called `avgSumOfSquares` that expects no arguments. Instead, this function gets its input from the user at the keyboard. The function asks the user to enter a series of numbers, one at a time. The user types `end` to indicate that there are no more numbers. The function computes the average of the sum of the squares of all the values entered. For example, given the values 6, -3, 4, 2, 11, 1, and -9, the sum of the squares would be  $(36 + 9 + 16 + 4 + 121 + 1 + 81) = 268$ . The average of the sum of squares would then be  $268/7 = 38.285714285714285$ . The function then prints the average of the sum of the squares and returns that average. However, if `end` is entered before any values are entered, the function notifies the user that no numbers were entered and returns `None`. You may assume the user inputs are valid: they will either be a number or the string `end`. Here are some examples of how your function should behave:

```
>>> avgSumOfSquares()
Enter next number: 6
Enter next number: -3
Enter next number: 4
Enter next number: 2
Enter next number: 11
Enter next number: 1
Enter next number: -9
Enter next number: end
The average of the sum of the squares is: 38.285714285714285
38.285714285714285
>>> avgSumOfSquares()
Enter next number: 3.27
Enter next number: -1.9
Enter next number: 6
Enter next number: -1
Enter next number: end
The average of the sum of the squares is: 12.825725
12.825725
>>> avgSumOfSquares()
Enter next number: end
No numbers were entered.
>>> x = avgSumOfSquares()
Enter next number: end
No numbers were entered.
>>> print(x)
None
>>> x = avgSumOfSquares()
Enter next number: -1
Enter next number: 2
Enter next number: -3
Enter next number: end
The average of the sum of the squares is: 4.666666666666667
>>> print(x)
4.666666666666667
```

### **Where to do the assignment**

You can do this assignment on your own computer, or in the labs. In either case, use the IDLE development environment -- that's what we'll use when we grade your program. Put all the functions you created in a file called "prog3.py".

### **Submitting the Assignment**

Go to Canvas and submit the file containing your functions as an attachment. Do NOT cut-and-paste your functions into a text window. Do NOT hand in a screenshot of your functions' output. We want one file from you: "prog3.py".

### **Saving your work**

If you are working in the lab, you will need to copy your program to your own flash-drive. To save it on flash-drive, plug the flash-drive into the computer (your TA or the staff in the labs can help you figure out how), open the flash-drive, and copy your work to it by moving the folder with your files from the Desktop onto the flash-drive.