Introduction to programming – ASSIGNMENTS 6

General instructions:

- Type in and test your programs using Python Idle.
- Bring in your answers to the next assignment (i.e. demonstration) session in memory stick, or save them to a web folder accessible in class. Alternatively, you can bring written answers with you, though this is not recommended.
- Remember to comment your code – this does NOT mean, that every single line should be commented. Comment the important parts in your program.
- Prepare to present your solution to the class.

1.

Write a program that queries the user for numbers one at a time, and appends all the numbers entered into a list. When the user enters a zero, the program outputs the list and then terminates.

Example run:

```
Enter a number or a zero to quit: 3
Enter a number or a zero to quit: 7
Enter a number or a zero to quit: 1
Enter a number or a zero to quit: 0
List: [3, 7, 1]
```

2.

Write a function `average(m),` which calculates and returns the average of all items in the list **m**.

Example run (in Python Shell):

```
>>> myList = [2,4,6,8,10]
>>> print average(myList)
6.0
```

3.

Write a function `maximumValue(myList)` which returns the largest of the values stored in the list **myList**.

Example run (in Python Shell):

```
>>> n = [1,5,3,1,4,2,3]
>>> print maximumValue(n)
5
```

4.

Write a function `evenNumbers(myList)`, which returns a new list with all the even numbers stored in the **myList**. The function should not change **myList** in any way.

Example run (in Python Shell):

```
>>> n = range(1,11)
>>> otherList = evenNumbers(n)
>>> print otherList
[2, 4, 6, 8, 10]
```

5.

Write a function `mergeLists(list1, list2)`, which gets two sorted lists as a parameter, joins these lists into a new list and then return's the merged list. The merged list should also be sorted in increasing order.

Example run (in Python Shell):

```
>>> n = [1, 3, 5, 7, 9]
>>> m = [2, 4, 8, 12]
>>> newList = mergeLists(n, m)
>>> print newList
[1, 2, 3, 4, 5, 7, 8, 9, 12]
```

6. _** Expert assignment (double points)_

Write a procedure removeDuplicates(n), which finds and removes duplicate items from the list given as an argument. Hence, after the procedure is called, only a single instance of each item can be found in a list. Do not change the order of items!

Example run:

```
>>> myList = [1, 2, 1, 3, 3, 2, -1, 5, 3, 5, -1, 2, 5]
>>> removeDuplicates[myList]
>>> print myList
[1, 2, 3, -1,  5]
```