CS 260
Spring 2015
Lab 5

- Download and modify the Python3 program named "Lab5.py" so that it simulates the quicksort algorithm as described below.
- The program should display the stack of recursive calls that are active immediately after each call, and also immediately after each return, of the recursive quicksort function.
- Each function call will push a new activation onto the stack, and similarly each return will pop an activation from the stack.
- Each activation includes the values of the parameters (left, right) that are passed to the quicksort function.
- The program should behave as shown in the following example runs, but note that the output varies with each run due to the use of a random number generator for choosing the pivot position.
- Here are two example runs using the same input length = 10.

```
Enter length of list: 10
Tracing stack of recursive calls for Quicksort:
Call:     [(0, 9)]
Call:     [(0, 9), (0, 2)]
Call:     [(0, 9), (0, 2), (0, 0)]
Return:   [(0, 9), (0, 2)]
Call:     [(0, 9), (0, 2), (2, 2)]
Return:   [(0, 9), (0, 2)]
Return:   [(0, 9)]
Call:     [(0, 9), (4, 9)]
Call:     [(0, 9), (4, 9), (4, 7)]
Call:     [(0, 9), (4, 9), (4, 7), (4, 5)]
Call:     [(0, 9), (4, 9), (4, 7), (4, 5), (4, 4)]
Return:   [(0, 9), (4, 9), (4, 7), (4, 5)]
Call:     [(0, 9), (4, 9), (4, 7), (4, 5), (6, 5)]
Return:   [(0, 9), (4, 9), (4, 7), (4, 5)]
Return:   [(0, 9), (4, 9), (4, 7)]
Call:     [(0, 9), (4, 9), (4, 7), (7, 7)]
Return:   [(0, 9), (4, 9), (4, 7)]
Return:   [(0, 9), (4, 9)]
Call:     [(0, 9), (4, 9), (9, 9)]
Return:   [(0, 9), (4, 9)]
Return:   [(0, 9)]
Return:   [ ]
```

```
Enter length of list: 10
Tracing stack of recursive calls for Quicksort:
Call:     [(0, 9)]
Call:     [(0, 9), (0, 5)]
Call:     [(0, 9), (0, 5), (0, -1)]
Return:   [(0, 9), (0, 5)]
Call:     [(0, 9), (0, 5), (1, 5)]
Call:     [(0, 9), (0, 5), (1, 5), (1, 2)]
Call:     [(0, 9), (0, 5), (1, 5), (1, 2), (1, 0)]
Return:   [(0, 9), (0, 5), (1, 5), (1, 2)]
Call:     [(0, 9), (0, 5), (1, 5), (1, 2), (2, 2)]
Return:   [(0, 9), (0, 5), (1, 5), (1, 2)]
Return:   [(0, 9), (0, 5), (1, 5)]
Call:     [(0, 9), (0, 5), (1, 5), (4, 5)]
Call:     [(0, 9), (0, 5), (1, 5), (4, 5), (4, 3)]
Return:   [(0, 9), (0, 5), (1, 5), (4, 5)]
Call:     [(0, 9), (0, 5), (1, 5), (4, 5), (5, 5)]
Return:   [(0, 9), (0, 5), (1, 5), (4, 5)]
Return:   [(0, 9), (0, 5), (1, 5)]
Return:   [(0, 9), (0, 5)]
Return:   [(0, 9)]
Call:     [(0, 9), (7, 9)]
Call:     [(0, 9), (7, 9), (7, 7)]
Return:   [(0, 9), (7, 9)]
Call:     [(0, 9), (7, 9), (9, 9)]
Return:   [(0, 9), (7, 9)]
Return:   [(0, 9)]
Return:   [ ]
```

- When you are finished, upload your modified "Lab5.py" program to Blackboard.