5. Homework

Programming portion (problems 1 and 3) due **10/8/13** at 11:55pm on Blackboard.
Written portion (problem 2) due **10/9/13** at the beginning of class.

Please create a separate Python file for problem 1 and problem 3 below, and use the following naming convention: `lastName_firstName_hw5_number.py`.

In order to receive any credit for the programming portions, you are required to thoroughly comment and test your code.

1. **Sorted (7 points)**

   (a) (3 points) Write a function `is_sorted_incr` that takes as input a list of numbers, and returns `True` if the list is sorted in increasing order, and `False` otherwise. Test your function with several inputs.

   (b) (2 points) Write a function `is_sorted` that takes as input a list of numbers, and returns `True` if the list is sorted (either in increasing order or in decreasing order), and `False` otherwise. Test your function with several inputs.

   (c) (2 points) What is the asymptotic running time of the functions in terms of `n`, where `n` denotes the length of the input lists? Please write your answer as a comment in your code, together with a very brief justification.

2. **Mystery (5 points)**

```
def mystery(x):
    if len(x)==0 or len(x)==1:
        return x
    y = mystery(x[1:])
    y.append(x[0])
    return y
```

   (a) (1 point) What types are valid for the input variable `x`?

   (b) (3 points) On paper, trace the execution of `mystery` with an input of size 3. Draw the recursion stack and show how it changes. The trace should result in the final returned output.

   (c) (1) What does `mystery` do? Please describe in words the functionality of the `mystery` function.

3. **Recursion (6 points)**

   (a) (3 points) Write a **recursive** function `min_rec(L)` that computes the minimum of a list `L` of numbers. Add comments to your code to point out the base case and the recursive case. Test your code.

   (b) (3 points) Write a **recursive** function `sum_even(n)` that computes the sum of all positive even numbers less than the input parameter `n`. Add comments to your code to point out the base case and the recursive case. Test your code.