Homework on Recursion

1. Given a string, compute recursively a new string where all the 'x' chars have been removed.

   Example:
   Input: noX("xaxb") Output: "ab"
   Input: noX("abc") Output: "abc"
   Input: noX("xx") Output: ""

2. Given a non-negative integer n, return the sum of its digits recursively (no loops). Note that, mod (%) by 10 yields the rightmost digit (126%10 = 6), while divide (/) by 10 removes the rightmost digit (126/10 = 12).

   Example:
   Input: sumDigits(126)
   Output: 9
   Input: sumDigits(49)
   Output: 13
   Input: sumDigits(12)
   Output: 3

3. Given a string and a non-empty substring sub, compute recursively if at least n copies of sub appear in the string somewhere, possibly with overlapping. N will be non-negative.

   Example:
   Input: strCopies("catcowcat", "cat", 2)
   Output: True
   Input: strCopies("catcowcat", "cow", 2)
   Output: False
   Input: strCopies("catcowcat", "cow", 1)
   Output: True

4. Write a method/function called findMax_recursive( ) that takes the head of a Singly LinkedList in its parameter and returns the maximum number from the linkedlist. You may use helper functions.