Programming Assignment #4

Recursion practice

CS 3358.253, Spring 2015
Instructor: Jill Seaman

Due: Wednesday, 4/1/2015 (upload electronic copy by 1:30pm)

Problem:

Write functions for each of the following problems. Each problem should be solved by writing a recursive function (potentially with an auxiliary/driver function). Your final program should not have any loops in it.

All of your solutions should be in a single .cpp file. The main function of the file should demonstrate each of your solutions, by running some tests and producing some output.

1. **Table of Squares:**
   Convert the following function to one that uses recursion. Note that the values in the output table are **increasing** (from 1 to n, not n down to 1).

```
void tableOfSquares (int n) {
   for (int num=1; num<=n; num++) {
      cout << num << "      " << (num * num) << endl;
   }
}
```

2. **Recursive Power Function**
   Write a function **power** that uses recursion to raise a number to a power. The function should take two arguments, the number to be raised to the power (floating point) and the power (a non-negative int).

3. **isMember function**
   Write a boolean function named **isMember** that takes three arguments: an array of ints, its size (number of elements) and a target value. It should return true if the target value is found in the array, or false if the value is not found in the array.

4. **maxNode function**
   Write a function **maxNode** that takes a pointer to the first node of a linked list composed of nodes of the following type:

```
struct Node {
  int  value;
  Node *next;
};
```

The function should return the largest value in the list. Your function should fail if the list is empty. (Put the definition of the Node struct at the top of your file). See the Linked List (NumberList) demo on the website for a file called ClasslessLinkedList.cpp for an example of coding a linked list without a class. In the main function, you will need to create a linked list to pass to this function for testing.

5. **Palindrome detector**
   A palindrome is any word, phrase, or sentence that reads the same forwards or backwards. Here are some palindromes (find more with google):

   level
   Pot top
   A man a plan a canal Panama

   Write a boolean function that determines if a string argument is a palindrome. The function should return true if the argument reads the same forwards and backwards. For full credit, your function should ignore spaces and be case-insensitive. Assume the input is just letters and spaces. Hint: you can do recursive calls on **any** sub-string of the original one (as long as it is shorter, and you have enough base cases).

Sample Output:

```
Table of squares:
N      N Squared
  1            1
  2            4
  3            9
  4           16
  5           25
  6           36
  7           49
  8           64
  9           81
 10          100

Power function:
2 to the 5th power: 32

Member function
1 2 3 4 5 has 4?: YES
1 2 3 4 5 has 6?: NO

MaxNode of {10,101,11,9,100): 101

Is "A man A plan A canal Panama" a palindrome? YES
Is "No one" a palindrome? NO
```

NOTES:

- Just one file. 5 (or more) functions plus main for testing.
- No static variables!
- No global variables or tables!
- No loops!! No for loops, no while loops, no do-while loops.

Style:

See the Style Guidelines document on the course website.

Logistics:

Please submit your solution in a single file. You can call it assign4_xxxxxx.cpp.

The xxxxx is your TX State NetID (your txstate.edu email id).

Submit: an electronic copy only, using the Assignments tool on the TRACS website for this class.