CS213M: Assignment 1

Problem 1: Singly linked list

Due Date: 25/01/2015

We are providing you with files named **`node.hpp`** and `node.cpp` to start with. They contain the definition of a class **`Node`**, with two member variables - a next pointer, and a data item (an integer to be stored in the list node). You also will be provided with a file **`list.hpp`** with declarations for functions you have to define in a new file named **`list.cpp`**. The declarations of the functions you need to define are reproduced below for quick reference.

Note: Do not modify the given code. Do not change the signatures of the functions below. Do not use C++ STL.

Functions to be defined

Note: Assume that the input lists for the functions below do not have any cycles. Assume that you would not be given a 'bad' `pos` in the `insert` or `delete` functions i.e. `pos` will be a non-negative integer less than the size of the input list.

1. `Node* insert(Node* head, int val, int pos);`

   Inserts a new node with **`val`** contained in it at the position **`pos`** and returns the head of the new list. We number the nodes starting from 0. Thus if the existing list is 1 -> 2 -> 3 -> 4 and one calls **`insert`** on it with **`pos`** being 3 and **`val`** being 7, the list becomes 1 -> 2 -> 3 -> 7 -> 4.

2. `Node* remove(Node* head, int pos);`

   Remove the node at the position **`pos`**. We number the nodes starting from 0. Thus if the existing list is 11 -> 12 -> 13 -> 14 and one calls **`remove`** on it with **`pos`** being 3, the final list becomes 11 -> 12 -> 13.

3. `int size(Node* head);`

   Gets the size of the linked list.

4. `Node* reverse(Node* head);`

   Reverses a list. Returns the head of the new list.

5. `int cycle_size(Node* head);`

   A singly linked list can contain a cycle (Can you see how?). We now want to determine the length of the cycle in a list. If the list is acyclic, return 0. You have to write a function to compute this length given a list. This function is included in the declarations given in `list.hpp`.

IMPORTANT NOTE: A **NULL** value for the head of a list denotes an empty list.

Don't forget to read the comments in the given header file.