CS 330 Intro to the Design and Analysis of Algorithms

Homework 1

(20 pts)

Use the following Node definition for problems 1 - 4:

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

1. Write a function to concatenate two linked lists. Given lists l1 = (2, 3, 1) and l2 = (4, 5), after return from concatenate(l1, l2) the list l1 should be changed to be l1 = (2, 3, 1, 4, 5). Your function should not change l2 and should not directly link nodes from l1 to l2 (i.e. the nodes inserted into l1 should be copies of the nodes from l2.)

```
void concatenateList(Node*& h1, Node* h2 );
//
// Precondition: h1 and h2 are head pointers of linked lists.
// The lists may be empty or non-empty.
//
// Postcondition: A copy of list h2 is concatenated (added to the end)
// of list h1.  List h2 should be unchanged by the function.
// NOTE: The nodes added to the list h1 must be copies of the
// nodes in list h2.
```

2. Write a function to insert a number as the new ith node of a linked list. Nodes initially in positions i, i+1, ..., n should be shifted to positions i+1, i+2, ..., n+1. Thus, the length of the list will increase by 1. If the original list contains fewer than i-1 nodes, then the number should be inserted at the end of the list.

3. Write a function to remove duplicate entries in a linked list. For example, given the list (5, 2, 2, 5, 3, 9, 2) as input, your function should change the list so that on return from the function it contains (5, 2, 3, 9).

4. Write a function that returns the maximum data value in a linked list. You should assume that the list contains non-negative integer values in the data fields of all nodes. Given a list list1 = (2, 3, 9, 1, 5), after execution of the statement

```
max = maxElem(list1);
```

the variable max should be equal to 9. If the list is empty, you should return the value -1.

Follow the **Submission Template** to organize your submission. Develop a **comprehensive test plan** that is all implementation ready. Your source code should be **fully documented** and conform to **good programming style**. Your submission should show that you have fully implemented your developed test plan.

Observed output must clearly follow implementation of the test plan – it should be clear that the output was generated by the source code provided and should align with the test plan. Please paste your complete source code at the end of the document.