

## **Programming Assignment I COP3502H**

**Due: 1/25/07 11:59pm to Web CT**

Welcome to the Computer Sciences Corporation. We are a startup company specializing in software tools for business professionals. Since we are in the early stages of our company, we have absolutely no support software for building our applications. As one of our first employees, it is your job to develop the fundamental data structures and algorithms we will need for further application development.

The first data structure library we will need is set of linked list routines. This library will be used in many of the data structures you develop in future assignments.

### **Requirements**

Use the following struct for you linked list routines.

```
struct node {
    int id;
    char name[50];
    double assets;
    struct node *next;
};
```

You must implement the following linked list routines.

1. AllocateNode – this function returns a pointer to new node structure.
2. PrependNode – add a node to the beginning of the list
3. AppendNode – add a node to the end of a list
4. InsertNode – given a node with a particular id number, insert a new node after it.
5. SearchList – given an id number, return the correct node.
6. PrintNode – given an id number, print the data in the correct node.
7. PrintList – print out the contents of each node in the list
8. DeleteFirstNode – delete the first node in the list
9. DeleteLastNode – delete the last node in the list
10. DeleteNode – given an id number, delete the node that contains it.
11. ReverseList – reverse nodes in the list (NOTE: TO GET FULL CREDIT YOU MUST REVERSE THE NODES IN PLACE)
12. ListUnion – given 2 linked list A and B, return their union based on id.
13. ListIntersection – given 2 linked lists A and B, return their intersection based on id.
14. FindMthToLast – return the mth-to-last node in the list (if m=0, the last element in the list is returned).

You should put the function prototypes in a header file “linkedlist.h” and use

“`linkedlist.c`” for your implementation. That way, you can simply include “`linkedlist.h`” in your `main.c` file for testing and in further assignments.

## **Documentation**

You must document your routines. Each function should be commented with what the function does, its input parameters, and what it returns. Comments are very important and will account for 15% of your grade.

## **Testing**

You must thoroughly test your routines to ensure that they work. You will need to develop a main routine to test them and output the test results into a file. These results should be easily readable. For each routine, think about the type of input you might have and develop a test case for it.

## **Deliverables**

You must submit all source files in addition to your test files to WebCT by the due date. Also, include a README file explaining how you tested your code and what problems you encountered. Note the code must be able to be compiled and executed on the Olympus unix system.

## **Grading**

Grading will be based on the following:

- 60% correct functionality
- 25% testing code
- 15% documentation