Computer Science II, Spring 2003
Exam 2, April 9
Howard and Townsend

Name: _____

If you need more space, please continue on the back of the page.

[HasCl questions omitted]

5. (20 points) Given the ADT List from our Chapter 3 with methods:

```
bool isEmpty( ) const;
int getLength( ) const;
void insert(int index, ListItemType newItem, bool &success);
void remove(int index, bool &success);
void retrieve(int index, ListItemType dataItem, bool &success) const;
```

Consider the case where `ListItemType` is `int`.

a) Write a function, using the methods above, to compute the average of the integers in a list.

b) Write a function, using the methods above, to bubble sort a list. You <u>may</u> <u>not</u> assume the swap that you wrote for homework. You may refer to the following bubble sort pseudocode for a list of size *theLength*, since bubble sort is Exam 1 and Final material.

x ← 1 to *theLength* – 1
$\quad\quad$ y ← 1 to *theLength* – 1
$\quad\quad\quad\quad$ If $item_y > item_{y+1}$ then swap the items

6. (20 points) Design a class to represent the ADT ExamGrade. Develop methods to setGrade( ), displayGrade( ), curveGrade( ). The details about this ADT and its methods are purposely sketchy, because we want you to DESIGN the ADT. The parameters, in particular, are missing. This much we know:

The exam is a normal exam, graded 0 to 100, inclusive.
The grade will be displayed as an A, B, C, D or F. No + or -. 90, 80, 70, 60 cutoffs.
The teacher may curve the exams by specifying one of the parameters to curveGrade as the amount to add to the grade; there may or may not be other parameters.

a) You are to supply the full C++ ADT with .h and .cpp files. Don't worry about includes. Don't include a test file.

b) You must follow the principles outlined in Chapter 1. One of these is "fail-safe" programming. You must account for user error in your code. If you do not do this, you will not receive full credit. You must also select one other principle from Chapter 1 on your own (without a hint, such as the one we just provided for you regarding "fail-safe" programming). You are to name this principle and discuss BRIEFLY how you applied it to your design and/or code.

7. (15 points) Short answer

a) Write code to declare a pointer named `p` that will point to an `int`.

b) Write code to obtain an `int` cell to which `p` will point. (assume there is an available cell in memory)

c) Write code to place a 3 in the `int` cell to which `p` points.

8. (9 points) Write a function that takes a pointer to the head of a linked list as its parameter and returns the largest integer in the linked list.

Assume the following structure definition for a node of the linked list:

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