CSC 210 (Spring, 1999) Sample Final Exam page 1 of 2

Note: For questions related to linked lists, assume the class List has a data member my_head and that class Node is defined as follows:

```cpp
class Node {
public:
    int my_data;
    Node * my_link;
    Node( int data, Node * link = 0 ){
        my_data( data ), my_link( link )
    }
    ~Node() { delete my_link; }
};
```

General Programming Skills.

For each of the following standard programming situations, write syntactically correct C++ code to accomplish the required task.

1. [5 points] Declare a pointer variable A and make it point to an allocated array of 100 integers that has all 0's in it.

2. [5 points] Make the variable s_new point to a freshly allocated C-string that is a copy of the string pointed to by s_old (assume both variables have been declared; you may use anything in string.h except std::). 

3. [5 points] Add the integer stored in the_int (previously declared) to the front of a linked list pointed to by my_head (just write the actual code, not the member function).

4. [5 points] Remove and deallocate the first node of a linked list pointed to by my_head.

5. [5 points] Read characters from standard input until a Q is encountered and store all (except the Q) as a C-string in an array buffer (assume buffer has been declared to hold 512 bytes; your code should not write beyond the end of the array; either truncate the input or abort instead).

Linked List Programming.

Each of these two problems should be solved without using any helper functions.

6. [12 points] Write a recursive function int min( Node * p ) that returns the minimum integer in the linked list pointed to by p. Your function should return INT_MAX (the maximum possible value for an int — assume this is already defined) if p is null.

7. [18 points] Write an iterative member function swap_last_two for the List class that reverses the order of the last two nodes of the list. Your function should not access the my_data field of any node. If the list has less than two nodes, it should do nothing.
Abstract Classes, Inheritance, and Virtual Functions.

8. [25 points] What does the following program print?

```cpp
#include<iostream.h>
#include<string.h>

class Base {
public:
    Base( const char * name )
    { cout << "Base::Base, " << name << endl;
        my_name = new char[ strlen( name ) + 1 ];
        strcpy( my_name, name );
    }
    void beta() { cout << "Base::beta" << endl; }
    virtual void gamma() = 0;
    virtual "Base();
    { cout << "Base::" << my_name << endl;
        delete [] my_name;
    }
protected:
    char * my_name;
};

class Zero: public Base {
public:
    Zero( const char * name, int number ):
        Base( name ), my_number( number )
    { cout << "Zero::Zero, " << number << endl;
    }
    void beta() { cout << "Zero::beta" << endl; }
    virtual void gamma() {
        cout << "Zero::gamma" << endl;
    }
    virtual "Zero();
    { cout << "Zero::" << my_number << endl;
    }
protected:
    int my_number;
};

class One: public Zero {
public:
    One( const char * name, int number, const char * occupation ):
        Zero( name, number )
    { cout << "One::One, " << occupation << endl;
        my_operation = new char[ strlen( occupation ) + 1 ];
        strcpy( my_operation, occupation );
    }
    void beta() {
        cout << "One::beta" << endl;
    }
    virtual void gamma() {
        cout << "One::gamma" << endl;
    }
    virtual "One();
    { cout << "One::" << endl;
        << my_operation << endl;
        delete [] my_operation;
    }
protected:
    char * my_operation;
};
```

Finite-State Machines.

9. [10 points] Design a finite-state machine for a program that reads input and prints all HTML commands, one command per line. For the purposes of this exercise, an HTML command is any string that appears between angle brackets <>. For example, if the input is

```
Here is some <strong>bold</strong> text,
and here is some <em>italics</em>.
```

the output should be

```
strong
/strong
em
/em
```

Do not write the actual program, but indicate the actions of the program on the transitions of the FSM.

EXTRA CREDIT.

10. [10 points] What are the phases of the “software life cycle”? Explain briefly what happens during each phase.