Sample CPSC301 Final Day 1-1

**Problem 1**. Complete the given C++ program (prob1.cpp) to read an array, print the array, find and print the smallest positive number. Basically, complete the two functions printArray and getElement.

* printArray must print the values in the array in order with a space and comma between each one and an endline after the last value.
* getElement must return the smallest positive number. If there are no positive numbers in the array it must return 0. Test these cases.

A sample run is given below:

Enter number of integers : 5

Enter 5 integers: 4 6 8 -12 -9

Contents of array: 4, 6, 8, -12, -9

Output of getElement: 4

A second test case:

Enter number of integers : 5

Enter 5 integers: -1 -2 -3 -4 -5

Contents of array : -1 -2 -3 -4 -5

Output of getElement: 0

**File submission:** Upload exactly one file to Titanium:

1. prob1.cpp

**Problem 2**. You are given a partially implementation of class IntegerLinkedList which stores integers in a singly linked list. Add a public member function that does the following:

* int getInteger (int i): return the integer stored at the i-th node of the linked list. I=0 corresponds to the first node (pointed to by the head pointer), i=1 corresponds to the second node, etc. If the list is empty (the head pointer is NULL), the return value should be -1.

To test your code, a main function is provided to you in prob2.cpp.

* You must implement all your code in the given header file called IntegerLinkedList.h
* Do not add other member variables and functions to the class IntegerLinkedList. (local variables inside the function are ok)
* You may change prob2.cpp for test purposes, but the class should still work with an unmodified prob2.cpp.
* Do not use any STL classes.

**File submission:** Upload exactly one file to Titanium:

IntegerLinkedList.h

**Problem 3**. Add a function int MaxList() to IntegerLinkedList.h that uses recursion to find the maximum value on the list. Add test code to main to test this function.

* You may change prob2.cpp for test purposes, but the class should still work with an unmodified prob2.cpp.
* Do not use any STL classes.
* HINT: You will need to add a helper function since head is not public!
* A non-recursive version gets no credit.

**File submission:** Upload exactly one file to Titanium:

1. IntegerLinkedList.h