Exam in Programming Proficiency (EPP) Part I

Published text will be allowed (open book policy), but no lecture notes, copies of personal program listings, or electronic media (hard drives, flash drives, cell phones, etc.) will be allowed except for authorized software (compiler, word processor, IDEs like Visual Studio, XCode) already installed on the lab computers. Furthermore, no access to the Web will be allowed during the exams except for accessing the Titanium site.

You might find yourself under some time pressure in this examination. Please check the point value for each problem so that you do not spend more time on one problem than it is worth.

Please make sure to read each problem carefully before working on it.

Please PRINT your name: _____

Please SIGN your name: _____

IMPORTANT:

- **Starter code is given to you on Titanium**
- Do not change filenames of starter code when submitting your code.
- Upload ONLY your C++ files (.cpp, .h) to Titanium. Do NOT upload any Visual Studio solution files.

COMMAND TO COMPILE AND RUN CODE ON LINUX/TUFFIX (RECOMMENDED):

```
clang++ -std=c++17 prob1.cpp
./a.out
```

INSTRUCTIONS TO GET STARTER CODE INTO VISUAL STUDIO:

1. Download starter code from Titanium.
2. Create new empty project
    - File → New → Project → Empty Project
    - Remember the folder where the project is created
3. Open the Visual Studio project folder using Windows Explorer
4. Move (drag & drop) starter code from Downloads to the Visual Studio folder
5. In Visual Studio, add “existing items”
    - Project → Add Existing Item … → select the .cpp/.h files
6. After solving the problem, upload the files in the Visual Studio folder to Titanium

Problem	Max	Earned
1	20	
2	20	
3	20	
Total	60	

Problem 1. Complete the given C++ program (prob1.cpp) to read an array of integers, print the array, and find the **2nd largest element in the array**. A main function in prob1.cpp is given to read in the array. You are to write two functions, `printArray`() and `getElement`(), which are called from the main function. `printArray`() outputs integers with a space in between numbers and a newline at the end. `getElement`() returns an integer that is the **second largest** element in the array. You may assume that the integers are unique.

A sample run is given below:

```
Enter number of integers: 5
Enter 5 integers: 23 45 10 17 92
Contents of array: 23 45 10 17 92
Output of getElement: 45
```

- Write only the two functions in the space provided.
- You can change the main function for your own testing. Your code will be tested with a similar main function.
- Do not change the name of the given file (it should remain prob1.cpp)
- Do NOT use any STL classes (such as std::vector).

File submission: Upload prob1.cpp with the two functions to Titanium

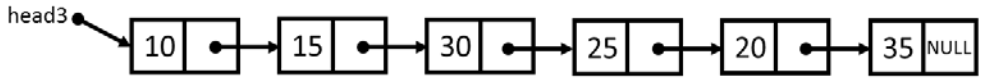
Problem 2. You are given the partial implementation of `class IntegerLinkedList` which stores integers in a singly linked list. Add a public member function that does the following:

`int getSum():` return the sum of all integers stored in the linked list.

A main file (`prob2.cpp`) is provided to you that will read integers from the user and call the public member functions of an object of your class.

A sample run is shown below (with the corresponding linked list):

```
Enter number of integers: 6
Enter 6 integers: 35 20 25 30 15 10
getSum: 135
```



- You must implement all your code in the given cpp file called `IntegerLinkedList.cpp`
- You cannot add other member variables and functions to class `IntegerLinkedList`.
- You can change the main function for your own testing. Your code will be tested with a similar main function.
- Do NOT use any STL classes (such as std::list, std::forward_list).

File submission: Upload exactly one file:

1. `IntegerLinkedList.cpp`

Command to compile and run on Linux/Tuffix (RECOMMENDED):

```
clang++ -std=c++17 prob2.cpp IntegerLinkedList.cpp
./a.out
```

Problem 3. Add a recursive function called `getSumRecurse` to `class IntegerLinkedList` to calculate the sum of the linked list's data values (same as in Problem 2 but recursive).

A recursion "helper" function is already included in `class IntegerLinkedList.` You only need to write the recursive function.

A sample run is given below:

```
Enter number of integers: 6
Enter 6 integers: 35 20 25 30 15 10
getSumRecurseHelper(): 135
```

- **A non-recursive version of the function will get no credit.** The function should not have any loops at all. Do **not** use any global variables.
- Write only the recursive function in IntegerLinkedList.cpp
- You can change the main function for your own testing. Your code will be tested with a similar main function.
- Do NOT use any STL classes (such as std::vector).

File submission: Upload exactly one file to Titanium:

1. IntegerLinkedList.cpp

Command to compile and run on Linux/Tuffix (RECOMMENDED):

```
clang++ -std=c++17 prob3.cpp IntegerLinkedList.cpp
./a.out
```

Note: The answers to both problem 1 and 2 will be in class IntegerLinkedList. If you have implemented both answers in one file, IntegerLinkedList.cpp, upload the same file to Titanium for BOTH assignments.