Department of Computer Science
Graduate Exam in Programming
Fall 2016

You must write **Java** or **C++** on the first page of your answers!! You may answer the questions using either Java or C++, but you must use the same language for each question on the exam.

1. **(2 pts)** Define a function `sumDigits` that takes a positive integer as an argument and returns the sum of the digits in the number. For example, if the argument is 7823, the result of `sumDigits(7823)` would be 20. Do not convert the integer to string (do not use `itoa()` or `toString()`, use arithmetic operations instead).

2. **(1.5 pts)** Write a **recursive** boolean function named `isMember` that takes three arguments: an array of strings, its size and a target string. It should return true if the target is found in the array, or false if the target is not found in the array. Do not assume that the array is sorted.

   **Do not use loops, extra parameters, or global or static variables.**

3. **(2.5 pts)** Implement the following two functions that operate over arrays of int (assume the values in the array are unique):

   A. `indexOfMax(array, size)`: This function should return the index of the largest element of the array (assume that the size is greater than 0).

   B. `selectionSort(array, size)`: This function should sort the elements of the array using the following algorithm: find the largest element and move it into the last position, then find the next largest element and move it into the second to last position, and so on, until the entire array is sorted. Do not use any other algorithm to sort the list. This function should call the `indexOfMax` function from part A.
4. \( (4 \text{ pts}) \) Declare and implement a class to represent a **stack** of integers, called **IntStack**. Include the following functions in your class:

- a default constructor that creates an empty stack.
- a void function for push(x) that inserts new value onto the top of the stack.
- an int function for pop() that removes the value from the top of the stack, and returns it (if the stack is empty, return -1).
- a boolean function isempty() that returns true if the stack is empty, otherwise false.

Note that a stack adds and removes elements in “Last In, First Out” order. The most recently added element is the one that is removed next.

Implement the class functions using a **linked list** composed of Nodes to store the values in the stack. Use the following declarations in your class:

```cpp
// C++
private:
    struct Node {
        int value;
        Node *next;
    };
```

```java
// Java
private class Node {
    int value;
    Node next;
}
```