# Assignment 7

| | |
|---|---|
| Minimum/Maximum Templates | 25 pts. |
| Template Iterative Binary Search | 25 pts. |
| search function | 20 pts. |
| deleteDuplicates function | 40 pts. |
| **TOTAL** | **110 pts.** |

## *Part 1 (Templates)*

### Minimum/Maximum Templates

Write templates for the two functions `minimum` and `maximum`. The `minimum` function should accept two arguments and return the value of the argument that is lesser of the two. The `maximum` function should accept two arguments and return the value of the argument that is the greater of the two. Design a simple driver program (main()) that demonstrates the templates with various data types

### Template Iterative Binary Search

Write a template version of the iterative binary search (code provided). In the comments next to the template definition specify requirements on the template parameter type. Include additional code in main that tests your new template to work well not only for integers, but for characters too.

## *Part 2 (Linked Lists)*

### NumberList Functions

Modify the `NumberList` class (provided) to include two new member functions:

1. Function `search` should search the list for a specific value. If the value is found, function should return the number of the element in the list that holds the value. Start counting elements from 0. If the value is not found, the function should return -1. Demonstrate the function in a driver program.
   Function header: `int NumberList :: search(int key)`
   Example:
   Given the list 2 -> 7 -> 1 -> 7 -> 12 -> 7

   ```
   int n= list.search(1);//  n holds 2 now
    n= list.search(15);//  n holds -1 now
   ```

2. Function `deleteDuplicates` should scan the linked list and delete all duplicates of the given value in the list. Only one element with the given value should be left in the list. Do not assume that the list is sorted. If there was no duplicates of the given value (or no such value found), the list should stay unchanged.
   Function header: `void NumberList ::deleteDuplicates(int key)`
   Example:
   list: 2 -> 7 -> 1 -> 7 -> 12 -> 7
   function call: `list.deleteDuplicatest(7);`
   list after function call : 2 -> 7 -> 1 -> 12
   Demonstrate the function in a driver program.
   **Requirement:** Do not call `deleteNode` member function in your function implementation.