CST 238 – Fall 2013
Homework 5
Due: 10/25/2013 (11:55 PM)

1. (20 points) Download the sample dynamic list class (DList.cpp, DList.h, and listtester.cpp) from the textbook's web site

http://cs.calvin.edu/books/c++/ds/2e/SourcePrograms/Chap06/Figure6.2/

Make a project named **`hw5`** with the three files. Then, update the program to meet the following requirements.

(a) Develop a new function member called **`addItem()`** to add an item to the list. This function should keep all items in the List class in the ascending order after adding the new item. Note that several duplicated items are allowed in the **List** class. In the function, if the addition is not possible because the array is already full. In that case, your function should create a new dynamic array. But the new array's size should be twice (= double) of the current array's size. Then, your function should copy the content of the current array to the new array. After that, delete the current array and the variable "myArray" should point to the new array. The following presents the function prototype:

```
void addItem(ElementType item);
```

(b) Develop a new function member called **`deleteItem()`** to remove an item(s) from the list. If the item to be removed has several duplications in the array, your function should remove all of them in the array. If your function delete at least one element in the array, it should return true. However, the function doesn't delete any element because the item doesn't exist in the array, it should return false. The following presents the function prototype:

```
bool deleteItem(ElementType item);
```

(c) Develop a new function member called **`findLast()`** that returns the position (= index) of a specified item in the array. If the array has several duplications with the value, the function should return the last index among them. If the item doesn't exist, it should return -1.

```
int findLast(ElementType item);
```

The following file named **`listtester.cpp`** presents a sample test driver program. Demonstrate that your modified **`DList.h`** and **`DList.cpp`** work correctly with this test program. If the driver program has any error, let the instructor know it.

```
1.  //--- listtester.cpp: Program to test List class.

2.  #include <iostream>
3.  using namespace std;

4.  #include "DList.h"

5.  int main()
6.  {
7.     // Test the class constructor with two elements.
8.     List intList (2);
9.     cout << "Constructing intList\n";

10.    // Test addItem() by adding 30, 30, 40, 50, 10, and 60.
11.    cout << "Adding 30." << endl;
```

```
12.     intList.addItem(30);
13.
14.     cout << "Adding 30." << endl;
15.     intList.addItem (30);

16.     cout << "Adding 40." << endl;
17.     intList.addItem (40);
18.     cout << intList << endl;

19.     cout << "Adding 50." << endl;
20.     intList.addItem (50);

21.     cout << "Adding 10." << endl;
22.     intList.addItem (10);

23.     cout << "Adding 60." << endl;
24.     intList.addItem (60);
25.
26.     // Display the content of the array
27.     cout << intList << endl;

28.     // Test findLast()
29.     cout << "Position (30): " << intList.findLast(30) << endl;
30.     cout << "Position (70): " << intList.findLast(70) << endl;


31.     // Test deleteItem() by deleting 30, 40, and 60 from the list
32.     cout << "deleteItem 30: " << intList.deleteItem(30) << endl;
33.     cout << "deleteItem 40: " << intList.deleteItem(40) << endl;
34.     cout << "deleteItem 60: " << intList.deleteItem(60) << endl;
35.     cout << intList << endl;

36.     return 0;
37. }
```

How to turn in?

Submit your source programs (**DList.cpp and DList.h**) on iLearn.