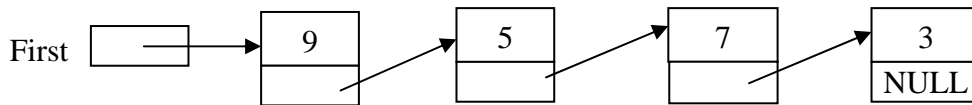


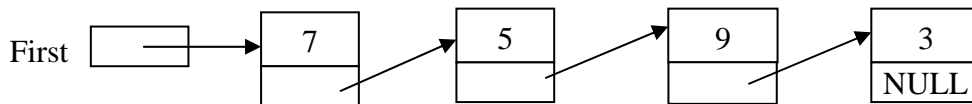
**CST 238 – Fall 2013**  
**Homework 6**  
**Due: 04/01/2013 (11:55 PM)**

1. (20 points) Download the sample code of a linked list class (**Linked List Class – Sample**) from the **iLearn**. Make a project called **hw6** with the three files.

- (a) Develop a new function member called **swap()** to swap two elements in the list. For example, let's assume that the list has the following diagram:



Then, the result of the **swap(0, 2)** will be like this



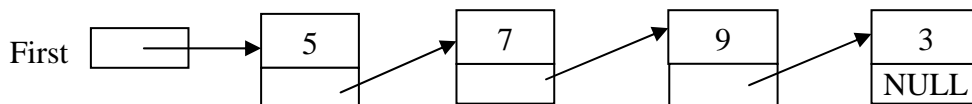
In the diagram, the function swaps the content of the first element (= 9) with the content of the second element (= 7).

The following presents the function prototype.

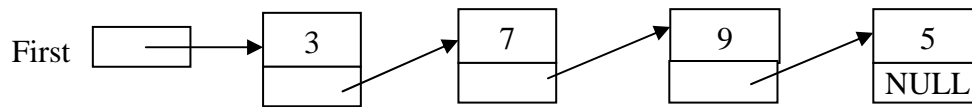
**bool swap(int first\_index, int second\_index);**

If the function can't be executed by any reason (for example, an index is not valid.), the function should return false. Otherwise, it should return true.

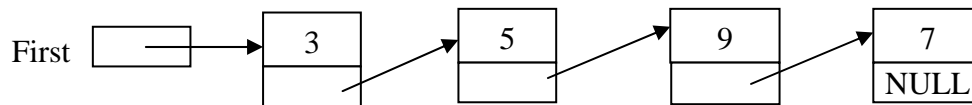
- (b) Develop a new function member called **ascendSort()** to sort the list in the ascending order. For this, you will need to use the function **swap()**. For instance, let's assume that the list has the following diagram:



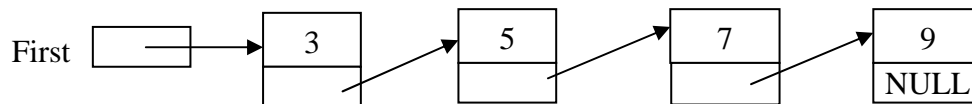
Now, your function should find out the smallest element (3 in this case) in the list and swap the element with the first element (5 in the case). So the result would be like below:



Since you found the smallest element in the list, you should identify the next smallest number in the remaining list (5 in this case) and swap it with the second element (7 in the case).



By continuing this step on the remaining list, you can get the sorted result as below.



This is the function prototype.

```
void ascendSort();
```

The following program, **linked\_list\_tester.cpp**, presents a sample driver program. If there's a problem in the driver program, let the instructor know.

```

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

3.  #include "LinkedList.h"

4.  int main()
5.  {
6.      LinkedList intList;
7.
8.      intList.insert(200, 0);
9.      intList.insert(100, 0);
10.     cout << "Test swap(0,1): " << intList.swap(0,1) << endl;
11.     intList.display(cout);
12.
13.     intList.insert(300, 2);
14.     intList.insert(400, 1);
15.     intList.insert(500, 3);
16.     cout << "Test swap(1,4): " << intList.swap(1,4) << endl;
  
```

```
17.     cout << "Test swap(6,4): " << intList.swap(6,4) << endl;
18.     intList.display(cout);
19.     cout << endl;
20.
21.     intList.ascendSort();
22.     intList.display(cout);
23.     cout << endl;
24.
25.     return 0;
26. }
```

### **How to turn in?**

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