Lab Exercises

a) maxItem() function

i) List with many elements:

This screenshot shows a list that was created with multiple elements and the maxItem() function being used to determine the largest of the elements.

```
11
12 #include "LinkedList.h"
13
14 int main()
15 {
16    // Test the class constructor
17    LinkedList intList;
18    cout << "Constructing intList\n";
19
20    // Test insert()
21    intList.insert(300, 0);
22    intList.display(cout);
23    cout << endl;
24
25    intList.insert(200, 0);
26    intList.display(cout);
27    cout << endl;
28
29    intList.insert(100, 0);
30    intList.display(cout);
31    cout << endl;
32
33    intList.insert(400, 3);
34    intList.display(cout);
35    cout << endl;
36
37    intList.insert(800, 4);
38    intList.display(cout);
39    cout << endl;
40
41    //Test maxItem
42    cout << "\nMax is "<< intList.maxItem()<< endl;
43
```

Console

<terminated> (exit value: 0) hw1 - labLinkedLists (1) [C/C++ Application] /User

```
Constructing intList
300
200 300
100 200 300
100 200 300 400
100 200 300 400 800

Max is 800
```

ii) List with 1 element

In this screenshot the list contains one element and the maxItem() function determined that single element to be the element with the highest value.

```
14 int main()
15 {
16    // Test the class constructor
17    LinkedList intList;
18    cout << "Constructing intList\n";
19
20    // Test insert()
21    intList.insert(300, 0);
22    intList.display(cout);
23    cout << endl;
24
25    //Test maxItem
26    cout << "\nMax is "<< intList.maxItem()<< endl;
27
28 //   //Test isAscendingOrder
29 //   cout << "\nIs it ascending order?: " << intList.isAscendingOrder()<< endl;
30
31    // Test destructor
32    {
33       LinkedList anotherList;
```

Console

<terminated> (exit value: 0) hw1 - labLinkedLists (1) [C/C++ Application] /Users/Carlos/eclipse-workspace-cpp/hw1 -

```
Constructing intList
300

Max is 300
```

iii) Empty List

In this screenshot, the maxItem() function shows -1 as the max because the list is empty and prints the “error -- no list” message.

```
14 int main()
15 {
16    // Test the class constructor
17    LinkedList intList;
18    cout << "Constructing intList\n";
19
20    //Test maxItem
21    cout << "\nMax is "<< intList.maxItem()<< endl;
22
23 //   //Test isAscendingOrder
24 //   cout << "\nIs it ascending order?: " << intList.isAscendingOrder()<< endl;
25
26    // Test destructor
27    {
28       LinkedList anotherList;
29       for (int i = 0; i < 10; i++)
30       {
31     anotherList.insert(100*i, i);
32       }
33       cout << "\nThis is another list\n";
```

Console

<terminated> (exit value: 0) hw1 - labLinkedLists (1) [C/C++ Application] /Users/Carlos/eclipse-workspace-cpp/hw1 - labLi

```
Constructing intList

Max is -1

This is another list
0 100 200 300 400 500 600 700 800 900

Two items are erased from the first list

Error -- no list
Illegal location to delete -- 1
Illegal location to delete -- 1
```

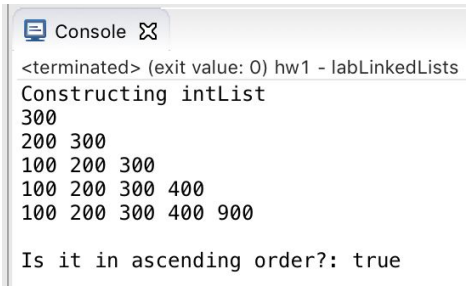
b) isAscendingOrder() function

i) List with many elements

These screenshots show the inAscendingOrder() function working when the list is and isn't in ascending order and contains multiple elements.

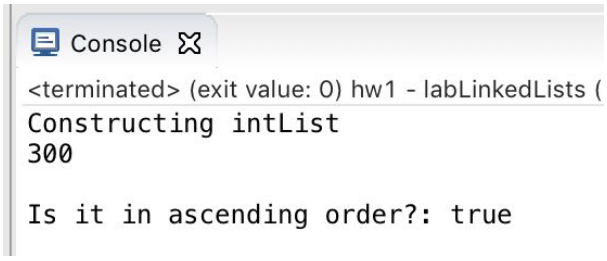
```
Constructing intList
300
200 300
100 200 300
100 200 300 400
100 200 300 400 300

Is it in ascending order?: false
```



ii) List with one element

This screenshot shows the isAscendingOrder() function determining the list is in ascending order when it contains only one element.



iii) Empty list

This screenshot shows the isAscendingOrder() function returning true when a list has been constructed but nothing has been inserted into it and it is therefore empty.