Practical Lab 06 – Linked Lists

1. Create a Java program that will incorporate the `LinkedList` and `Iterator` classes from the lecture notes (these are located in the code folder) and produce the following output when executed:

```
1.  Linked List Example!
2.  Linked list data: 11 22 33 44
3.  Add 55 to the beginning of the Linked List
4.  Now the list contains: 55 11 22 33 44
5.  Add 66 to the end of the Linked List
6.  Now the list contain: 55 11 22 33 44 66
7.  Add 99 at the 3rd location in the List
8.  Now the list contain: 55 11 99 22 33 44 66
9.  Display the first element on the list (55)
10. Display the final element on the list (66)
11. Display the element at the 4th position on the list (22)
12. Remove the element at the beginning of the list: 55
13. Now the list contains: 11 99 22 33 44 66
14. Remove the element at the end of the list: 66
15. Now the list contain: 11 99 22 33 44
16. Remove the element at the 2nd position on the list: 99
17. Now the list contain: 11 22 33 44
```

2. Think about where you could return the list size, write a method (called `listLength()`) that would do this and allow the size of the linked list to be returned after an item has been added or removed from the list.

3. What would you need to do to implement a doubly linked list, given the singly linked list that accompanied the lecture?
   Write pseudcode for a `previous()` method and a `hasPrevious()` method that would help implement a doubly linked list. What other changes to the `LinkedList` class and the `ListIterator` class would need to be made to successfully implement a doubly Linked List (think about the pointers that would allow you to iterate down through a list). Using the notes and the accompanying resources in the Week 7 Linked List folder along with any other resources you can research online.

4. Write a method that will allow a user to input a value and will search through the link list to find it. If found the method should add 100 to the value and insert it after the found value. If the value is not found the method should inform the user of this fact.