CSI 220 Data Structures and Algorithms
Fall 2022
Lab 2: Linked Lists

Using the linked list that we created in class as the base, complete the following:

Incorporate the following methods into your new class:

1. Method void insertFirst(T val) that will insert the value val at the start of the list.
2. Method void insertLast(T val) that will insert the value val at the end of the list
3. Method void insertIndex(T val, int inx) that will insert the value val in the index inx. If called insertIndex(5,1), it should insert the node with value 5 in position 1 (start). If called insertIndex(5,2), it should insert the node with value 5 in position 2 (one after the start). Be aware of potential errors!
4. Method void removeFirst() that will remove the value at the start of the list.
5. Method void removeLast() that will remove the value at the end of the list
6. Method void removeIndex(int inx) that will remove the value in position inx. If called removeIndex(1), it should remove the node in position 1 (Start). Be aware of potential errors!
7. Method int searchByValue(T val) that will return the index of the first instance of a value val in the list. For example if the list l = [1,5,7,5,2] and you call searchByValue(5), the method should return 1. Be aware of potential errors!
8. Method T searchByIndex(int inx) that will return the value of the node in the list in position inx (starting in 1). For example if the list l = [1,5,7,5,2] and you call searchByIndex(4), the method should return 5. Be aware of potential errors!
9. Method bool Update(int inx, T newVal) that will update the value of the node in position inx with newVal. The method will return true if it was successful in updating the value or false otherwise.
10. Method int UpdateAll(T oldVal, T new Val, int times) that will update the value of all nodes that have the value oldVal with newVal. The method will return the number of updates it completed (zero if none). The value *times* will determine how many nodes need to be updated in case that there are more than one nodes with the same value.
11. Method int UpdateAll(T oldVal, T newVal) that will update the value of all nodes that have the value oldVal with newVal. The method will return the number of updates it completed (zero if none).
12. Method String toString() that will return the list values as a String in order.
13. (Bonus) InsertSorted(T val) that will insert a node with the value val following increasing order. This will only affect the node to be inserted. For example, if the list has [5,9,2] and you insert 7 the list will be [5,7,9,2].
14. (BONUS + 5 points) ReSort() that will sort a list that is already created. You are free to use any sorting method

Each method should be properly commented and formatted. Include a Tester class that should call each method from your program at least twice. Use print statements to show that the changes are affecting the list. **For methods that could lead to errors (i.e. removing an element that does not exist) make sure that you test one successful and one unsuccessful case. Before you submit your document, comment out the unsuccessful cases, but leave them in your file.**