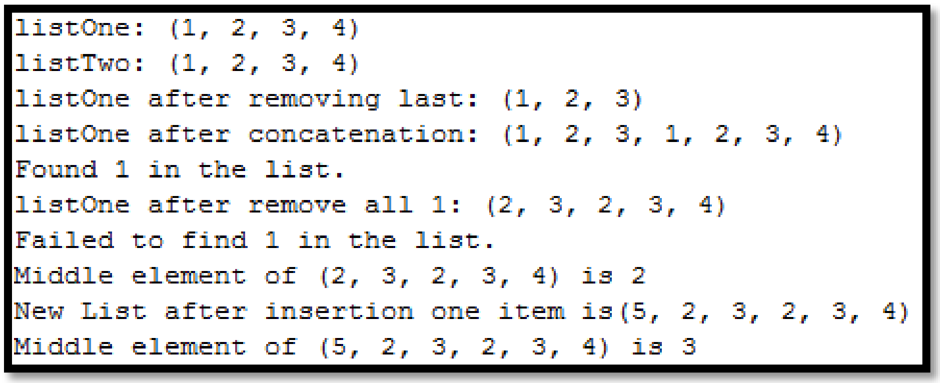
CSCI3200 Spring 2019 LAB2

Objective: The objective of this lab is to enhance the students' understanding on linked list based implementations.

1. Download the compressed folder LAB2 from D2L.
2. MyLinkedList is a linked list with two reference variables, where head always refers to the head of myLinkedList and tail always refers to end (last node if existed) of myLinkedList.
3. Based on the provided skeleton implement the following methods
    a. ***public int size():*** `this method will return the number of nodes on the linkedList`
    b. ***public boolean isEmpty():*** check if the list is empty or not
    c. ***public void addFirst(E e):*** //add a node to the front of the list
    d. ***public void addLast(E e):***//add a node to the end of the list
    e. ***public E removeFirst():***//remove the first node of the list and return the removed item
    f. ***public void removeLast( ):*** this method will remove the last node from the list if exist.
    g. ***public void concatenateList (***myLinkedList ***M)*** : this method will concatenate another myLinkedList referred as M at the end of this myLinkedList without creating any new node. For example, if you have this list is head->1->2->3->4<-tail, M: head->5->6<-tail then after call this method, you will have head->1->2->3->4->5->6<-tail.
    h. ***public int searchElement (E targetElement):*** this method will search if a targetElement is on a list or not, and return the occurrence of the target element.
    i. ***public void removeElement(E targetElement):*** this method will remove ***all*** occurrence of a targetElement from a list.
    j. ***Public E middleElement():*** This method will return the element store in the middle of the list. For example, if you have this list is head->1->2->3->4<-tail, by calling this method, it will return 2 or 3. If you have a list contains head->1->2->3->4->5<-tail, by calling this method, it will return 3.

Remarks: Please do not add other instance fields or methods. Do not change the provided method signature including return type. Only based on the given information to implement each method.

You can simply use the provided mainClass to test your code. The expected output is:



```
listOne: (1, 2, 3, 4)
listTwo: (1, 2, 3, 4)
listOne after removing last: (1, 2, 3)
listOne after concatenation: (1, 2, 3, 1, 2, 3, 4)
Found 1 in the list.
listOne after remove all 1: (2, 3, 2, 3, 4)
Failed to find 1 in the list.
Middle element of (2, 3, 2, 3, 4) is 2
New List after insertion one item is(5, 2, 3, 2, 3, 4)
Middle element of (5, 2, 3, 2, 3, 4) is 3
```