Linked Lists

---

1. Write a function `moveOdditemstoback()` that moves all odd integers to the back of a linked list.

   The function prototype
   ```
   def moveOdditemstoback(head):
   ```

   Sample Input and Output:
   Example 1:
   ```
   Enter a list of numbers, terminated by any non-digit
   character: 1 2 3 4 5 6 7 8 9 a
   Before: 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> 8 -> 9 -> None
   After: 2 -> 4 -> 6 -> 8 -> 1 -> 3 -> 5 -> 7 -> 9 -> None
   ```

   Example 2:
   ```
   Enter a list of numbers, terminated by any non-digit
   character: 2 7 18 3 4 15 b
   Before: 2 -> 7 -> 18 -> 3 -> 4 -> 15 -> None
   After: 2 -> 18 -> 4 -> 7 -> 3 -> 15 -> None
   ```

   Example 3:
   ```
   Enter a list of numbers, terminated by any non-digit
   character: 2 4 6 8 10 c
   Before: 2 -> 4 -> 6 -> 8 -> 10 -> None
   After: 2 -> 4 -> 6 -> 8 -> 10 -> None
   ```

2. Write the function `removeNode()` using the LinkedList structure defined in the lecture materials. The function should remove a node at the specified index from the linked list.

   The function definition is as follows:
   ```
   def removeNode(ll, index):
   ```

   Sample output:
   ```
   Enter one number per line (press Enter after each number).
   Enter any non-digit character to finish input:
   10
   20
   30
   40
   50
   a
   Current List has 5 elements: 10 20 30 40 50
   Enter the index of the node to be removed: 2
   After the removal operation:
   Current List has 4 elements: 10 20 40 50
   Enter the index of the node to be removed: 0
   After the removal operation:
   Current List has 3 elements: 20 40 50
   Enter the index of the node to be removed: 5
   Index out of range
   The node cannot be removed.
   Current List has 3 elements: 20 40 50
   ```

3. Write a function `split()` that copies the contents of a linked list into two other linked lists.

   The function prototype is given below:

```
def split(ll):
```

The function should copy nodes with even indices (0, 2, 4, etc.) to evenList and nodes with odd indices (1, 3, 5, etc.) to oddList. The original linked list should remain unmodified.

Sample output:

```
Enter one number per line (press Enter after each number).
Enter any non-digit character to finish input:
10
Successfully inserted 10 at index 0
20
Successfully inserted 20 at index 1
30
Successfully inserted 30 at index 2
40
Successfully inserted 40 at index 3
50
Successfully inserted 50 at index 4
q

Before split() is called:
Current list: 10 20 30 40 50

After split() was called:
Current list: 10 20 30 40 50
Even list: 10 30 50
Odd list: 20 40
```

4. Write a function `duplicateReverse()` that creates a duplicate of a linked list with the nodes stored in reverse.

   The function prototype is given below:

   ```
   def duplicateReverse(ll):
   ```

   The function should return a new LinkedList object containing the reversed duplicate of the input linked list. If the input list is empty, return an empty LinkedList.

   Sample output:

   ```
   Enter a list of numbers, terminated by any non-digit
   character:
   1
   Successfully inserted 1 at index 0
   3
   Successfully inserted 3 at index 1
   5
   Successfully inserted 5 at index 2
   2
   Successfully inserted 2 at index 3
   4
   Successfully inserted 4 at index 4
   6
   Successfully inserted 6 at index 5
   a

   Before duplicateReverse() is called:
   The original list: 1 3 5 2 4 6

   After duplicateReverse() was called:
   The original list: 1 3 5 2 4 6
   The duplicated reverse list: 6 4 2 5 3 11
   ```