Tutorial – Linked Lists

Information: Program templates for questions are available in NTULearn. You must use them to implement your functions.

1. **(moveEvenItemsToBackLL)** Write a C function `moveEvenItemsToBackLL()` that moves all the even integers to the back of the linked list.

   The function prototype is given as follows:

   ```
   void moveEvenItemsToBackLL(LinkedList *ll);
   ```

   Some sample inputs and outputs sessions are given below:

   If the linked list is **2, 3, 4, 7, 15, 18**:

   ```
   The resulting Linked List after moving even integers to the
   back of the Linked List is: 3 7 15 2 4 18
   ```

   If the linked list is **2, 7, 18, 3, 4, 15**:

   ```
   The resulting Linked List after moving even integers to the
   back of the Linked List is: 7 3 15 2 18 4
   ```

   If the current linked list is **1, 3, 5**:

   ```
   The resulting Linked List after moving even integers to the
   back of the Linked List is: 1 3 5
   ```

   ```
   If the current linked list is 2 4 6:
   The resulting Linked List after moving even integers to the
   back of the Linked List is: 2 4 6
   ```

2. **(moveMaxToFront)** Write a C function `moveMaxToFront()` that traverses a linked list of integers at most once, then moves the node with the largest stored value to the front of the list.

   The function prototype is given as follows:

   ```
   int moveMaxToFront(ListNode **ptrHead);
   ```

   For example, if the linked list is (**30, 20, 40, 70, 50**), the resulting linked list will be (**70, 30, 20, 40, 50**).

   ```
   1: Insert an integer to the linked list:
   2: Move the node with the largest stored value to the front of the list:
   0: Quit:

   Please input your choice(1/2/0): 1
   Input an integer that you want to add to the linked list: 30
   The Linked List is: 30
   Please input your choice(1/2/0): 1
   ```

```
Input an integer that you want to add to the linked list: 20
The Linked List is: 30 20
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 40
The Linked List is: 30 20 40
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 70
The Linked List is: 30 20 40 70
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 50
The Linked List is: 30 20 40 70 50

Please input your choice(1/2/0): 2
The resulting Linked List is: 70 30 20 40 50

Please input your choice(1/2/0): 0
```

3. **(removeDuplicatesSortedLL)** Write a C function `removeDuplicatesSortedLL()` that removes all duplicate values from a sorted linked list. ***You may assume that the list is already in ascending sorted order.***

The function prototype is given below:

```
void removeDuplicatesSortedLL(LinkedList *ll);
```

For example:

If the linked list is (**1, 2, 2, 4, 4, 5, 5**), the resulting linked list will be (**1, 2, 4, 5**).
If the linked list is (**1, 2, 3, 4, 5**), the resulting linked list will be (**1, 2, 3, 4, 5**)

Sample test cases are given below:

```
1: Insert an integer to the linked list:
2: Remove duplicates from a sorted linked list:
0: Quit:

Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 1
The resulting linked list is: 1
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 2
The resulting linked list is: 1 2
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 2
The resulting linked list is: 1 2 2
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 4
The resulting linked list is: 1 2 2 4
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 4
The resulting linked list is: 1 2 2 4 4
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 5
The resulting linked list is: 1 2 2 4 4 5
Please input your choice(1/2/0): 1
Input an integer that you want to add to the linked list: 5
```

```
The resulting linked list is: 1 2 2 4 4 5 5

Please input your choice(1/2/0): 2
The resulting linked list after removing duplicate values from the sorted
linked list is: 1 2 4 5

Please input your choice(1/2/0): 0
```

4. **(appendLL)** Write a C function `appendLL()` that takes two lists, ‘a’ and ‘b’, appends ‘b’ onto the end of ‘a’, and then sets ‘b’ to NULL (since it is now trailing off the end of ‘a’).

The function prototype is given below:

```
void appendLL(LinkedList *ll_a , LinkedList *ll_b);
```

For example, assume that two linked lists are as follows:

```
Linked list a: 1 2 3 4 5
Linked list a: 6 7 8 9 10

The resulting linked list a: 1 2 3 4 5 6 7 8 9 10
The resulting linked list b: Empty
```

Sample test cases are given below:

```
1: Insert an integer to the linked list a:
2: Insert an integer to the linked list b:
3: Merge two sorted linked lists:
0: Quit:

Please input your choice(1/2/3/0): 1
Input an integer that you want to add to the linked list a: 1
The resulting linked list a: 1
Please input your choice(1/2/3/0): 1
Input an integer that you want to add to the linked list a: 2
The resulting linked list a: 1 2
Please input your choice(1/2/3/0): 1
Input an integer that you want to add to the linked list a: 3
The resulting linked list a: 1 2 3
Please input your choice(1/2/3/0): 1
Input an integer that you want to add to the linked list a: 4
The resulting linked list a: 1 2 3 4
Please input your choice(1/2/3/0): 1
Input an integer that you want to add to the linked list a: 5
The resulting linked list a: 1 2 3 4 5
Please input your choice(1/2/3/0): 2
Input an integer that you want to add to the linked list b: 6
The resulting linked list b: 6
Please input your choice(1/2/3/0): 2
Input an integer that you want to add to the linked list b: 7
The resulting linked list b: 6 7
Please input your choice(1/2/3/0): 2
Input an integer that you want to add to the linked list b: 8
The resulting linked list b: 6 7 8
Please input your choice(1/2/3/0): 2
```

```
Input an integer that you want to add to the linked list b: 9
The resulting linked list b: 6 7 8 9
Please input your choice(1/2/3/0): 2
Input an integer that you want to add to the linked list b: 10
The resulting linked list b: 6 7 8 9 10

Please input your choice(1/2/3/0): 3
The resulting linked lists after appending list b to list a are:
The resulting linked list a: 1 2 3 4 5 6 7 8 9 10
The resulting linked list b: Empty

Please input your choice(1/2/3/0): 0
```