Assignment-1

You need to write a class called LinkedList that implements the following List operations:

1. **public int size();**

// returns the number of items in the list

2. **public void add(int index, Object item);**

// adds an item to the list at the given index, that index may be at start, end or after or before the
// specific element

3. **public void remove(int index);**

// removes the item from the list that has the given index

4. **public void remove(Object item);**

// finds the item from list and removes that item from the list

5. **public List duplicate();**

// creates a duplicate of the list

// postcondition: returns a copy of the linked list

6. **public List duplicateReversed();**

// creates a duplicate of the list with the nodes in reverse order

// postcondition: returns a copy of the linked list with the nodes in

7. **public List ReverseDisplay();**
//print list in reverse order

8. **public Delete_Smallest();**
// Delete smallest element from linked list

9. **public List Delete_duplicate();**
// Delete duplicate elements from a given linked list.Retain the earliest entries.

10. public List Print_middle();
// Find the middle element of a given linked list. In case of tie print the second one. Input:

    5 -> 7 -> NULL
    5 -> 7 -> 17 -> NULL
    5 -> 7 -> 17 -> 13 -> NULL
    5 -> 7 -> 17 -> 13 -> 11 -> NULL
    Output:7 7 17 17

11. public List Palindrome();
//Check whether a given singly linked list is palindrome or not.
Input:

    a -> b-> NULL
    a -> b -> a-> NULL
    s -> a -> g -> a -> r-> NULL
    r -> a -> d -> a -> r-> NULL

Output:

    not palindrome
    palindrome
    not palindrome
    palindrome

12. public List frequency();
// Calculate the frequency of occurrence of each element in a given linked listin the same order
// they appear.Avoid printing multiple entries.
Input:

    20 -> 18 -> 15 -> 20 -> 6 -> 18 -> 5 -> 18 -> NULL

Output:

    Freq(20) =2
    Freq(18) =3
    Freq(15)=1
    Freq(6)=1
    Freq(5)=1

13. Create two circular linked lists and find their maximum numbers. Merge the two circular linked lists such that the maximum number of 2nd circular linked list immediately follows the maximum number of the 1st circular linked list.
Input:

    12 -> 28 -> 18 -> 25 -> 19-> NULL
    5 -> 24 -> 12 -> 6 -> 15-> NULL

Output:

    28 -> 24-> 25 -> 15 -> 19 -> 15->5-> 18 -> 25 -> 19->NULL

14. Given a pairlinked lists, insert nodes of second linked list into the first linked list at alternate positions.Assume that the first linked list has at least as many elements as the second.

Input:

```
1 -> 2 -> 3 -> NULL
4 ->5 -> NULL
```

Output:

```
1 -> 4-> 2 -> 5 -> 3 -> NULL
```

15. Swap k-th node from the beginning with k-th node from the end in a linked list.

Input:

```
k=11 -> 2 -> 3 ->NULL
k=21 -> 2 -> 3 ->NULL
k=31 -> 2 -> 3 ->NULL
```

Output:

```
3-> 2 -> 1->NULL
1-> 2 -> 3->NULL
3-> 2 -> 1->NULL
```

16. Make a function that adds a linked list to itself at the end.

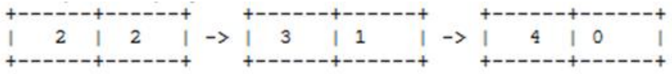
Input:

```
4 -> 2 -> 1 -> NULL
```

Output:

```
4 -> 2 -> 1 -> 4 -> 2 -> 1 -> NULL
```

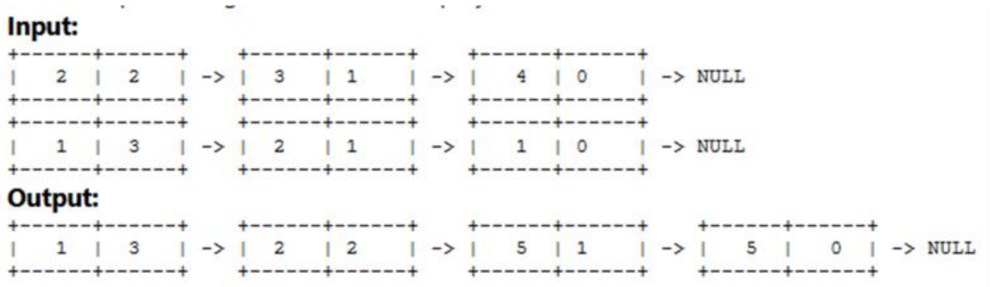
17. Write a program to represent a polynomial in variable X using a singly linked list, each node of which contains two kinds of data fields; one to store coefficient and another stores the power on variable X. For example, if a polynomial is: P(X)=2X^2 + 3X + 4 then the structure of linked list is as below:



```
+------+------+    +------+------+    +------+------+
|  2   |  2   | -> |  3   | 1    | -> |  4   | 0    |
+------+------+    +------+------+    +------+------+
```

Perform polynomial addition on two such polynomials represented by linked lists and generate a new list representing the sum of those polynomials.

Input:



Input:

```
+------+------+    +------+------+    +------+------+
|  2   |  2   | -> |  3   | 1    | -> |  4   | 0    | -> NULL
+------+------+    +------+------+    +------+------+
+------+------+    +------+------+    +------+------+
|  1   |  3   | -> |  2   | 1    | -> |  1   | 0    | -> NULL
+------+------+    +------+------+    +------+------+
```

Output:

```
+------+------+    +------+------+    +------+------+    +------+------+
|  1   |  3   | -> |  2   | 2    | -> |  5   | 1    | -> |  5   |  0   | -> NULL
+------+------+    +------+------+    +------+------+    +------+------+
```

Write the method to swap any two given nodes.

Input: swap(2, 4)

```
12 -> 28 -> 18 -> 25 -> 19-> NULL
```

Output

    12 -> 25 -> 18 -> 28 -> 19-> NULL

Note:

These nodes can be at different locations like::
First and last, Consecutive nodes, first and nth node, last and nth node etc..

18. Write the method to swap 1st node with last node and 2nd node with 2nd last node and so on. You have to swap addresses not values.

Input:

    28 -> 24-> 25 -> 15 -> 19 -> 17-> 18 -> 23 -> 19->NULL

Output:

    19 -> 23-> 18 -> 17 -> 19 -> 15-> 25 -> 24 -> 28->NULL

Note:

Don’t apply the reverse method

19. Reverse a linked list in groups of given size.

Input: size=4

    1->2->3->4->5->6->7->8->12->7->NULL

Output: 4->3->2->1->8->7->6->5->7->12->NULL

20. Perform pair-wise swapping of nodes of a given linked list.

Input:

    20 -> 18 -> 15 -> 10 -> 8 -> 6 -> 5 -> 3 -> 7 -> NULL

Output:

    18 -> 20 -> 10 -> 15 -> 6 -> 8 -> 3 -> 5 -> 7 -> NUL

Grading Criteria

Questions#	Marks	Total Marks
Question-1 to Question-10	1 mark each	10
Question-11 to Question-20	2 marks each	20

G☺☹D Luck