SENG201 DATA STRUCTURES AND ALGORITHMS I

Lab Assignment V

Prepared by Ms. Meltem Şirin

Write a program to check if the linked list is palindrome or not.

A palindromic list is the one which is equivalent to the reverse of itself.

To check whether a list is palindrome, first traverse list and check if any element from starting half doesn't match with any element from the ending half, then set the variable flag to false and break the loop. In the last, if the flag is false, then the list is palindrome otherwise not.

Algorithm and functions for this program should be as follows.

1. Create a Node which has two attributes: data and next. Next is a pointer to the next node in the list.

2. Create another node pointers represents head, tail.

3. Create Global variable for representing size.

4. **insert()** will add a new node to the list:

5. **reverseList()** will reverse the order of the node present in the list:

    a. Node current will represent a node from which a list needs to be reversed.

    b. Node prevNode represent the previous node to current and nextNode represent the node next to current.

    c. The list will be reversed by swapping the prevNode with nextNode for each node.

6. **isPalindrome()** will check whether given list is palindrome or not:

    a. Declare a node current which will initially point to head node.

    b. The variable flag will store a boolean value true.

    c. Calculate the mid-point of the list by dividing the size of the list by 2.

    d. Traverse through the list till current points to the middle node.

    e. Reverse the list after the middle node until the last node using reverseList(). This list will be the second half of the list.

    f. Now, compare nodes of first half and second half of the list.

    g. If any of the nodes don't match then, set a flag to false and break the loop.

    h. If the flag is true after the loop which denotes that list is a palindrome.

    i. If the flag is false, then the list is not a palindrome.

7. **display()** will display the nodes present in the list.