

DSC Practical – 6

Singly Linked List Operations and Applications.

1. Write algorithm and implement program for all operations of singly link list.
 - a. Creation of List
 - b. Inserting Node – as First Node, as Last Node, at desired location
 - c. Deleting Node – at First, at Last, Specific Node
 - d. Display List

2. Write algorithm for finding length of linked list via Iterative and Recursive method. Implement the same algorithm using 'C' program.

```
int getCount_Iterative(struct Node* head)
```

```
int getCount_Recursive(struct Node* head)
```

Note: Use the above singly linked list code (Definition 1) for the same.

3. Write algorithm and implement program to perform all stack operations using link list. *Implement PUSH, POP, PEEP, DISPLAY, PEEK [Peek operation gives you the TOP element but does not remove it from Stack]*

Note: Use the above singly linked list code (Definition 1) for the same.

4. Write algorithm and implement program to Sort a linked list.

sortList() will sort the nodes of the list in ascending order.

- Define a node current which will point to head.
- Define another node index which will point to node next to current.
- Compare data of current and index node. If current's data is greater than the index's data then, swap the data between them.
- Current will point to current.next and index will point to index.next.
- Continue this process until the entire list is sorted.

Note: Use the above singly linked list code (Definition 1) for the same.

5. Write algorithm and implement program to Reverse a linked list.
Initialize three pointers prev as NULL, curr as head and next as NULL.

Algorithm to reverse a Singly Linked List

%%Input : *head* node of the linked list

Begin:

```
    If (head != NULL) then
        prevNode ← head
        head ← head.next
        curNode ← head
        prevNode.next ← NULL
        While (head != NULL) do
            head ← head.next
            curNode.next ← prevNode
            prevNode ← curNode
            curNode ← head
        End while
        head ← prevNode
```

End if

End

Note: Use the above singly linked list code (Definition 1) for the same.