

## Practice Lab Assignment 5

### Practice Lab Assignment 5

For this Practice Lab Assignment, you will write programs in C implementing the concepts of **Linked Lists**.

#### Instructions

- There are 3 questions in this assignment.
- Any discussion with neighbor/or any other student is strictly not allowed.
- Mobile phones are not allowed. If found, disciplinary action may be taken.

**Due Date: This is only a Practice Lab so no submission is required.**

#### Grading Criteria

No Grading Criteria.

### Programming Questions

1. Create 3 files (sorted.h, sorted.c, test\_sorted.c) to implement a Sorted (in Increasing Order) Singly Linked List ADT (Implement only Insertion and Display Functions).

2. Write a C program that will remove all duplicate elements present in a singly linked list. You can use the below given insert() function in your program to test your code.

```
typedef struct node
{
    int data;
    struct node *next;
}node;
```

```
node *head=NULL;
```

```

void insert(int element)
{
    node *new = (node *)malloc(sizeof(node));
    new->data = element;
    new->next = NULL;
    if(head == NULL)
        head=new;
    else
    {
        node *temp=head;
        while(temp->next != NULL)
            temp = temp->next;
        temp->next = new;
    }
}

```

3. Create 3 different files (list.h, list.c, test\_list.c) to implement List ADT. Define 3 functions in it – insert(), swap() and display() as per below given specifications:

1. **Insert( )** function must take the element from the user and insert it in the linked list one after another (just think of “insert at the end” of the linked list).

2. **Swap( )** function must take 2 values (say X and Y) from the user and must swap the 2 respective nodes containing data X and Y. Make sure the 2 nodes must be swapped and not just the data of the node is swapped.

Your function must ensure the different positions of X and Y, specifically the following:

- (i) X and Y may or may not be adjacent.
- (ii) Either X or Y may be a head node.
- (iii) Either X or Y may be the last node.
- (iv) X and/or Y may not be present in the linked list.

**For Example :**

**Input:** 20->12->40->90->56 (Values to be swapped are 20 and 40)

**Output:** 40->12->20->90->56

3. **Display( )** function must display the current status of the linked list.