6) WAP to implement singly linked list with following operations.

a) Create a linked list.
b) Deletion of first element, specified element and last element in the list.
c) Display the contents of the linked list.

```
#include<stdio.h>
#include<stdlib.h>

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

struct node *head=NULL;

void create() {

    struct node *newnode;
    struct node *temp;
    int item;

    newnode=(struct node *) malloc (sizeof
                (struct node));
    printf("Enter the data \n");
    scanf("%d", &item);

    newnode->data = item;
    newnode->next = NULL;

    if(head == NULL) {
        head = newnode;
    }
```

```
else {
    temp = head;

    while (temp->next != NULL) {
        temp = temp->next;
    }

    temp->next = newnode;
}
}

void delete (int element) {

    struct node *temp, *del = NULL;

    temp = head;

    if (head = NULL) {
        printf ("Cannot delete. The list is
        empty\n");
    }

    if (head->data == element) {
        head = head->next;
        return;
    }

    while (temp->next != NULL) {
        if (temp->next->data == element) {
            del = temp->next;

            if (del->next = NULL) {
                temp->next = NULL;
            }
```

```
        else {
            temp->next = del->next;
        }
    }
    else {
        temp = temp->next;
    }
}

if (del == NULL) {
    printf("Element was not found. Please
            enter a valid element\n");
    }
}

void delFirst() {
    head = head->next;
    return;
}

void delLast() {
    struct node *temp, *del = NULL;

    temp = head;
    while (temp->next != NULL) {
        del = temp->next;

        while (del->next == NULL) {
            temp->next = NULL;
            return;
        }

        temp = temp->next;
    }
```

```
void display(){
    struct node *p;
    p=head;

    if(p=NULL){
     printf("There's no element/node
                in the list-\n");
    }
    else{
        while(p!=NULL){
          printf("%d\t", p->data);
          p=p->next;
        }
    }
}
```