1/2/24

LAB - 7

WAP to implement doubly linked list with primitive operations

(a) Create a doubly linked list.

(b) Insert a new node to the left of the node

(c) Delete the node based on specific value.

```
#include <stdio.h>
struct node
{
    int data;
    struct node *prev;
    struct node *next;
};
struct node *head = NULL;
void insert();
void delete();
void display();
void main()
{
    int ch;
    printf("Enter 1. Insert \n 2. Delete \n 3. Display
            4. Exit \n");
    while (ch != 4)
    {
        printf("Enter choice:");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1 : insert();
                     break;
```

```
            case 2: delete();
                    break;
            case 3: displayLL;
                    break;
        }
    }
    printf("Exited");
}

Void insert()
{   printf("Enter position: "); scanf("%d", &pos);
    struct node *new = (struct node*) malloc
                    (sizeof(struct node));
    printf("Enter the data:");
    scanf("%d", &new->data);
    if(head == NULL)
    {
        new->prev = NULL;
        new->next = NULL;
        head = new
        printf("Node inserted \n");
    }
    else { for(i=0; i<pos-1; i++)
    { ptr = ptr->next; }
    {   new->prev = ptr->prev;
        new->next = ptr;
    ptr->prev->next = new;
        ptr->prev = new;
        printf("Node inserted \n");
    }
}
```

```
void delete()
{
    int val;
    printf("Enter the value: ");
    scanf("%d", &val);
    struct node *ptr = head;
    if(head -> data == val)
    {
        head = ptr -> next;
        free(ptr);
        printf("Node deleted \n");
        return;
    }
    while (ptr -> data != val)
    {
        ptr = ptr -> next;
        if (ptr -> next == NULL)
        {
            ptr -> prev -> next = NULL;
            free(ptr);
            printf("Node deleted \n");
            return;
        }
    }
    ptr -> prev -> next = ptr -> next;
    ptr -> next -> prev = ptr -> prev;
    free(ptr);
    printf("Node deleted \n");
}
```

```
void display()
{
    struct node *p=head;
    while(p!=NULL)
    {
        printf("%d -> ",p->data);
        p=p->next;
    }
    printf("NULL\n");
}
```

OUTPUT :

```
Press 1. Insert
      2. Delete
      3. Display
Enter choice: 1
Enter data : 4
Enter choice: 1
Enter data : 5
Enter position: 1
Enter choice : 1
Enter position: 2
Enter data : 3
Enter choice : 3
5 -> 3 -> 4 -> NULL
Enter choice : 2
Node deleted
Enter choice : 3
5 -> 4 -> NULL
Enter choice : 4
Exited
```