Date: 05/02/24

1) WAP to implement doubly Link List with primitive operations

a) Create a doubly linked List

b) Insert new node to the left of the node

c) Delete the node based on specific value.

Soln

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

struct node{
        inte data;
        struct node *prev;
        struct node *next;
}
struct node *head = NULL;

void create_ll(){
        struct node *newnode, *ptr;
        int num;

        printf(" Enter -1 to exit );
        while(num != -1){
                printf("Enter the number");
                scanf("%d",&num);
                newnode = (struct node *)malloc(sizeof(struct node));
                newnode->data = num;
                if(head == NULL){
                        head = newnode;
                        newnode->next = NULL;
                } new-node->perr = NULL.
        else {

          ptr = head;

          while (ptr->next != NULL){

                ptr = ptr->next;};

          ptr->next = newnode;

          newnode->next = NULL;

           newnode->prev = ptr;

        }
      }
  }


void  insert_left(){

     struct node *newnode , *ptr; ~~preptr~~;

      int val; node;

      newnode = (struct node*) malloc (sizeof (struct node));

      ~~newnode->data=~~ val;

      printf("Enter the value:");

       scanf ("%d", &val);

      printf("Enter the value before which node has to be inserted)

      scanf ("%d", node);

      newnode->data = val;

       ptr = head;

       ~~while~~ while (ptr->data != node){

                    ptr = ptr->next;

        }

        ~~preptr = ptr->prev~~
  ~~prevptr -> next = ptr -> next~~
  ~~ptr -> next -> prev = prevptr~~

  newonode -> next = ptr

   newnode -> prev = ptr -> prev

    ptr -> prev -> next = newnode;

    ptr -> prev = newnode ;

}


void  display() {

   struct node *ptr;

    if (head == NULL){

          printf("Nothing to print");

    } else {

       ptr = head;

       while( ptr != NULL) {

             printf ("%d", ptr -> data);

             ptr = ptr -> next;

       }

   }


void   del-node () {

     struct node *ptr;

     int val;

     printf("enter the value to be deleted);

     scanf ("%d", val);

      if (head -> data = val){

          ptr = ptr -> next;

          ptr -> prev = NULL;

          head = ptr

          free(ptr);
else {
    while ( ptr -> data != val){
            ptr = ptr -> next;
        }
        ptr -> prev -> next = ptr -> next
        ptr -> next -> prev = ptr -> prev
        free (ptr);
    }
}
```

O/p:

```
---- MENU ----

1) Create ll
2) insert - left
3) delete
4) display
5) exit

Enter your choice : 1
Enter -1 to exit;

Enter the num: 10

Enter the num: 20

Enter the num: 30

Enter the num: 40

Enter the num: -1

Enter your choice : 4

    10 -> 20 -> 30 -> 40
```

```
Enter your choice : 2

Enter the value : 15

Enter a value before which data need to be Inserted: 20;

Enter your choice: 4

10 → 15 → 20 → 30 → 40

Enter your choice : 3

Enter the value to be deleted : 30

Enter your choice : 4

10 → 15 → 20 → 40.
```

5/2/24.