1) Implementation of a double linked list with insertion before a given element, deletion of given node.

=>

```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
struct node *next;
int data;
struct node *prev;
};
struct node *start =NULL;
struct node * create_ll (struct node *);
struct node* display(struct node*);
struct node * insert_before(struct node *);
struct node * delete_selected (struct node *);
int main()
{
int option;
do
{
    printf("\n **Main Menu**");
    printf("\n 1. create a list");
    printf("\n 2. Display list");
    printf("\n 3. Add a node before an element");
    printf("\n 4. Delete a given node");
    printf("\n 5. exit");
    printf("enter your option");
    switch(option)
    {
        case 1: start=create_ll(start);
                printf("\n Double_ll created");
                break;
```

```c
case 2 :
        start = display (start);
        break;
case 3 :
        start = insert_before (start);
        break;
case 4 :
        start = delete_selected (start);
        break;
  }
} while ((option) != 5);
   get ch ();
   return 0;
}
struct node * create_ll (struct node *start)
{
    struct node *new_node, *ptr;
    int num;
    pf ("\m entre -1 to exit");
    pf ("\m Enter the data");
    sf ("%d", &num);
    while (num! = -1)
    {  if (start == NULL)
        {  new node = (struct node *) malloc (sizeof (struct node));
           new node -> prev = NULL;
           new node -> data = num;
           new node -> next = NULL;
           start -> new node ;
        }
      else { ptr = start ;
        new node = (struct node *) malloc (sizeof (struct node *));
        new node -> data = num;
        while (ptr -> next != NULL)
               ptr = ptr -> next ;
        ptr -> next = new node ;
        new node -> prev = ptr ;
        new node -> next = NULL;
      } pf (" Enter data" );
        sf ("%d", &num); } return start ; }
```

```c
struct node * insert-before(struct
{
    struct node *newnode, *ptr;
    int num,val;
    pf("\n Enter the data:");
    sf("%d ",&num);
    pf("\n Enter the value before which the data has to be
                inserted");
    sf("%d "&val);
    newnode = (struct node *)malloc(sizeof(struct node));
        newnode->data=num;
        ptr=start;
        while (ptr->data != val)
            ptr=ptr->next;
    newnode->next=ptr;
    newnode->prev=ptr->prev;
    ptr->prev->next=newnode;
    ptr->prev=newnode;
    return start;
}
struct node display(struct node *start)
{
    struct node *ptr;
    ptr=start;
    while (ptr !=NULL)
    { pf("%d", ptr->data);
        ptr=ptr->next;
    }
    return start;
}
struct node delete_selected (struct node *start)
    { struct node *ptr;
        int val;
        ptr=start;
        pf("\n Enter the value to be deleted");
        sf("%d ",&val);
        while(ptr->data != val)
        { ptr=ptr->next;
        }
```

```c
if (ptr->data == val)
{
    ptr->prev->next = ptr->next;
    ptr->next->prev = ptr->prev;
    free(ptr);
}
else
{ printf("\n node with %d value doesnot exist \n", val);
}
return start;
}
```

o/p:
Main Menu:

1. Create a list
2. Display the list
3. Add a node before a given node
4. Delete a given node
5. Exit .

Enter your option : 1

    Enter -1 to end
      Enter the data : 23 45 67 -1

Enter your option:2
      23 45 6 7

Enter your option:3
    Enter the data: 6
    Enter the value before which the data has to be
    inserted : 45.

Enter your option : 4
    Enter the val to be deleted: 6.

Enter your option : 2
    23 45 6 45 7