

```

//Linked list of integers
#include<stdio.h>
#include<stdlib.h>

typedef struct node
{
    int data; //Node containing data
    struct node *next; //address part
}Node;

typedef struct
{
    Node *start;
}LL;

void append(LL *ln, int x)// to add nodes to a linked list
{
    Node *p;
    p = (Node*)malloc(sizeof(Node));
    p->data = x;
    p->next = NULL;

    if(ln->start == NULL)
    {
        ln->start = p;
    }
    else
    {
        Node *q;
        q = ln->start;
        while(q->next!=NULL)
        {
            q = q->next;
        }
        q->next = p;
    }
}

void insertbeg(LL *lptr, int ele) //to add node at beginning
{
    Node *p;
    p = (Node*)malloc(sizeof(Node));
    p->data = ele;
    p->next = lptr->start;
    lptr->start = p;
}

```

```

int countnodes(LL ln)
{
    int count = 0;
    Node *q;
    q = ln.start;

    while(q!=NULL)
    {
        count++;
        q = q->next;
    }
    return count;
}

void addAfterN(LL *lptr,int ele,int n)
{
    Node *q = lptr->start;
    Node *p;
    p = (Node*)malloc(sizeof(Node));
    p->data = ele;
    p->next = NULL;

    for(int i=1;i<n;i++)
    {
        q = q->next;
    }
    p->next = q->next;
    q->next = p;
}

void display(LL ln)// displaying the linked list
{
    Node *q;
    q = ln.start;

    printf("The elemets in the linked list are:\n");
    while(q!=NULL)
    {
        printf("%d\n",q->data);
        q = q->next;
    }
}

void deletenode(LL *lptr,int x)
{
    Node *q1,*p;
    q1 = lptr->start;

```

```

p = NULL;
while( q1 != NULL) //search the node
{

    if (q1->data == x) //data matches found, break while
        break;
    p = q1; //if not found, current node becomes previous node
    q1 = q1->next; //go to next node

}
if( q1 == NULL) //if desired node not found
{

    printf("Node with %d data not found\n",x);

}

else //if found

{

    if(lp1->start == q1) //if node to be deleted is first node
    {
        lp1->start = lp1->start->next; //or write q->next
        //update start, start will now point to second node
    }
    else
    {
        p->next = q1->next; //previous node points to node after node to be
deleted
    }
free(q1); //release the memory being pointed by q
}

}

void concat(LL *lp1, LL l2)
{

    Node *q;
    if (l2.start == NULL) //l2 is empty
        return;

    if(lp1->start==NULL) //l1 is empty and l2 is Not Empty
    {

        lp1->start=l2.start;

```

```

    }
    else
    {
        q=lp1->start; //locate last node of LL 1
        while(q->next != NULL)
        {
            q=q->next;
        }
        q->next=l2.start; //link last node of l1 with first node of l2
    }
}

void reverse(LL *lp1)
{
    Node *q,*r,*p;
    p=NULL; //previous node
    q=lp1->start; //current node
    while(q!=NULL)
    {
        r=q->next; //store next node in r pointer
        q->next=p; //current node points to previous node
        p=q; //for next iteration, current node becomes previous
        q=r; //for next iteration, r node becomes current node
    }
    lp1->start=p; //modify start so that it points to last node of original
LL
}

int main()
{
    LL l1,l2,l3; //three linked list of integers
    l1.start=NULL;
    l2.start=NULL;
    l3.start=NULL;
    int choice,ele;
    do
    {
        printf("\nEnter your choice : 1.Insert at beg LL1 2.Insert at end 3.Display
4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit\n");
        scanf("%d",&choice);
    }
}

```

```

switch(choice)
{
    case 1 :
        printf("Enter the element to be added at the beginning of the
linked list :");
        scanf("%d",&ele);
        insertbeg(&l1,ele);
        break;

    case 2 :
        printf("Enter the element to be added at the end of the linked
list :");
        scanf("%d",&ele);
        append(&l1,ele);
        break;

    case 3 :
        printf("The Linkedlist is :\n");
        display(l1);
        break;

    case 4 :
        printf("Enter the data to be deleted:");
        scanf("%d",&ele);
        deletenode(&l1,ele);
        break;

    case 5 :
        printf("The total nodes in the linked list are :
%d\n",countnodes(l1));
        break;

    case 6 :
        append(&l2,60);
        append(&l2,50);
        insertbeg(&l2,40);
        concat(&l1,l2);
        printf("The concatenated linked list is :\n");
        display(l1);
        break;
    case 7 :
        append(&l3,80);
        append(&l3,90);
        insertbeg(&l3,100);
        reverse(&l3);
        display(l3);

}

```

```

}

while (choice!=8);

return 0;
}

```

```

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
1
Enter the element to be added at the beginning of the linked list :10

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
1
Enter the element to be added at the beginning of the linked list :11

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
2
Enter the element to be added at the end of the linked list :12

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
3
The Linkedlist is :
The elemets in the linked list are:
11
10
12

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
4
Enter the data to be deleted:11

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
5
The total nodes in the linked list are : 2

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
6
The concatenated linked list is :
The elemets in the linked list are:
10
12
40
60
50

```

```

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
7
The elemets in the linked list are:
90
80
100

Enter your choice : 1.Insert at beg LL 1 2.Insert at end 3.Display 4.Delete Data 5.Count Nodes 6.Concat 2 LL 7.Reverse 8.Exit
8
PS C:\Users\Mark Lopes\Desktop\New folder (3)>

```