Linked List implication Code:-

```
#include<stdio.h>
#include<stdlib.h>
#include <conio.h>
typedef struct node
{
	int data; //node containing int data
	struct node *next;
}node;


typedef struct
{
	node *start;
}LL;

void createlist(LL *ll)
{
	node *p,*q;
	int arr[]={40,50,60,70,80};
	int i;
	for(i=0;i<5;i++)
	{
		p=(node *)malloc(sizeof(node));
		p->data=arr[i];
		p->next=NULL; //make the node as last node
		if(ll->start==NULL) //is this a first node?
		{
			ll->start=p;
		}
		else
		{
		q=ll->start;
			while(q ->next!=NULL)
			{
				q=q->next;
			}
		q->next=p;
		}
	}
}

void insert(LL *ll)
{
	node *p,*q;
	int i,c,j,k,c1;
	printf("Enter the No to be entered\n");
	scanf("%d",&i);
	p=(node *)malloc(sizeof(node));
	p->data=i;
	p->next=NULL;
	printf("Enter the location to enter the node\n");
```

```
	printf("1 for begining\n-1 for last node\n0 for node in between\n");
	scanf("%d",&c);
	if(c==1)
	{
		p->next= ll->start;
		ll->start=p;
	}
	else if(c==-1)
	{
		if(ll->start==NULL)
		{
			ll->start=p;
		}
		else
		{
			q=ll->start;
			 while(q ->next!=NULL)
			 {
				q=q->next;
			 }
			q->next=p;
		}
	}
	else if(c==0)
	{
		printf("Enter the index after which node is to be appended\n");
		scanf("%d",&j);
		q=ll->start;
		c1=0;
		for(k=0;k<j;k++)
		{
			q=q->next;
			if(q==NULL&&k<=j-1)
			{
				c1=1;
				break;
			}
		}
		if(c1==1)
		{
			printf("Index does not exist\n");
			return;
		}
		else if(c1==0)
		{
			p->next=q->next;
			q->next=p;
		}
	}

}

void display(LL *ll)
```

```
{
        node *p;
        int i=0;
        p=ll->start;
        printf("data index\n");
        while(p!=NULL)
        {
         printf("%d     %d\n",p->data,i++);
         p=p->next;
        }
}


void delete(LL *ll)
{
        int i,a;
        node *p,*q;
        printf("Enter the Value to be deleted\n");
        scanf("%d",&a);
    p = ll->start;
         q = NULL;
        while( p != NULL)
        {
                if (p->data == a)
                {
                break;
                }
                q = p;
                p = p->next;
        }
        if( p == NULL)
        {
        printf("Node with %d data not found\n",a);
        return;
        }
        else if(ll->start == p)
        {
        ll->start = p->next;
         }
        else
        {
        q->next = p->next;
        }
        printf("Node With data %d is deleted",a);
         free(p);
}

void count(LL *ll)
{
        int i=0;
        node *p;
        p= ll->start;
        while(p!= NULL)
```

```
	{
		i++;
	 p = p->next;
	}
	printf("The Total Number of Nodes in Linked List are %d",i);
}


void concat(LL *ll1,LL *ll2)
{
	node *p;
	printf("Linked list 1 before Concatintion\n");
	display(ll1);
	printf("Linked list 2 before Concatination\n");
	display(ll2);
	if (ll2->start == NULL)
	{
		return;
	}
	if(ll1->start==NULL)
	{
		ll1->start=ll2->start;
	}
	else
	{
		p=ll1->start;
		while(p->next != NULL)
		{
			p=p->next;
		}
		p->next=ll2->start;
	}
	printf("Linked list after Concatination\n");
	display(ll1);
}


void reverse(LL *ll)
{
	node *p,*n,*q;
	printf("Linked list before Reversing\n");
	display(ll);
	q=NULL;
	p=ll->start;
	while(p!=NULL)
	{
		n=p->next;
		p->next=q;
		q=p;
		p=n;
	}
	ll->start=q;
	printf("Linked list after Reversing\n");
```

```
        display(ll);
}


int main()
{
LL ll1,ll2,ll3;
ll1.start=NULL;
ll2.start=NULL;
ll3.start=NULL;
int choice,ele,c1;
createlist(&ll2);
createlist(&ll3);
while(1)
{
 printf("\nEnter your choice :\n1.Insert a node\n2.Display LL \n3.Delete Data \n4.Count Nodes in LL\n5.Concat 2 LL
\n6.Reverse LL\n7.ClearScreen \n8.Exit\n");
 scanf("%d",&choice);
 switch(choice)
 {
 case 1 : {insert(&ll1); break;}
 case 2 : {display(&ll1);break;}
 case 3 : {delete(&ll1); break;}
 case 4 : {count(&ll1);break;}
 case 5 : {concat(&ll1,&ll2); break;}
 case 6 : {reverse(&ll3);break;}
 case 7 : {clrscr();break;}
 case 8 : {printf("Thank You");exit(0);}
 default: {printf("Enter a valid Option\n");break;}
 }
}
return 0;
}
```

Output:-

```
Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
1
Enter the No to be entered
5
Enter the location to enter the node
1 for begining
-1 for last node
0 for node in between
1

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
1
Enter the No to be entered
6
Enter the location to enter the node
1 for begining
-1 for last node
0 for node in between
-1

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
1
Enter the No to be entered
8
Enter the location to enter the node
1 for begining
-1 for last node
```

0 for node in between
-1

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
1
Enter the No to be entered
7
Enter the location to enter the node
1 for begining
-1 for last node
0 for node in between
0
Enter the index after which node is to be appended
1

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
2
data index
5 0
6 1
7 2
8 3

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
3
Enter the Value to be deleted
8
Node With data 8 is deleted
Enter your choice :

1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
4
The Total Number of Nodes in Linked List are 3
Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
5
Linked list 1 before Concatintion
data index
5 0
6 1
7 2
Linked list 2 before Concatination
data index
40 0
50 1
60 2
70 3
80 4
Linked list after Concatination
data index
5 0
6 1
7 2
40 3
50 4
60 5
70 6
80 7

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
6

Linked list before Reversing
data index
40 0
50 1
60 2
70 3
80 4
Linked list after Reversing
data index
80 0
70 1
60 2
50 3
40 4

Enter your choice :
1.Insert a node
2.Display LL
3.Delete Data
4.Count Nodes in LL
5.Concat 2 LL
6.Reverse LL
7.ClearScreen
8.Exit
8
Thank You