Source Code

```
#include<iostream>
using namespace std;
class node{
public:
    int data;
    node* prev,*next;
    node(int val){
        data=val;
        prev=NULL;
        next=NULL;
    }
};
void insertAtHead(node* &head,int val){
    node* n=new node(val);
    if(head==NULL){
        head=n;
        return;
    }
    head->prev=n;
    n->next=head;
    head=n;
}
void insertAtTail(node* head,int val){
    if(head==NULL){
        insertAtHead(head,val);
    }
    node* n=new node(val);
    node* temp=head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    temp->next=n;
    n->prev=temp;
}
void insertAftLoc(node* &head,int key,int val){  //INSERT AFTER GIVEN K
EY
    if(head==NULL){
        return;
    }
    if(head->data==key){
        insertAtHead(head,val);
        return;
    }
    node* n=new node(val);
    node* temp=head;
    while(temp!=NULL and temp->data!=key){
        temp=temp->next;
    }
```

```
    if(temp==NULL){
        insertAtHead(head,val);
        return;
    }
    else if(temp->next==NULL){
        insertAtTail(head,val);
        return;
    }
    node* newnode=temp->next;
    temp->next=n;
    n->prev=temp;
    newnode->prev=n;
    n->next=newnode;
}
void deleteAtstart(node* &head){
    if(head==NULL){
        cout<<"List is empty"<<endl;
        return;
    }
    node* temp=head;
    temp->next->prev=NULL;
    head=temp->next;
    delete temp;
}
void deletion(node* &head,int val){  //delete At loc
    if(head==NULL){
        cout<<"List is empty"<<endl;
        return;
    }
    if(head->data==val){
        deleteAtstart(head);
    }
    else{
        node* temp=head;
        while(temp!=NULL and temp->data!=val){
            temp=temp->next;
        }
        if(temp==NULL){
            return;
        }
        temp->prev->next=temp->next;
        if(temp->next!=NULL){
            temp->next->prev=temp->prev;
        }
        delete temp;
    }
}
void display(node* head){
```

```
    cout<<"Display in original form: ";
    if(head==NULL){
        return;
    }
    while(head!=NULL){
        cout<<head->data<<"->";
        head=head->next;
    }
    cout<<"NULL"<<endl;
}
void display2(node* head){
    cout<<"Display in reverse: ";
    if(head==NULL){
        return;
    }
    while(head->next!=NULL){
        head=head->next;
    }
    while(head!=NULL){
        cout<<head->data<<"->";
        head=head->prev;
    }
    cout<<"NULL"<<endl;
}
int main(){

node* head=NULL;

insertAtHead(head,3);
insertAtHead(head,2);
insertAtHead(head,1);
display(head);
display2(head);
```

```
insertAftLoc(head,2,4);
insertAftLoc(head,4,5);
insertAftLoc(head,3,6);
insertAftLoc(head,8,7);
display(head);
```

```
insertAtTail(head,9);
insertAtTail(head,8);
display(head);
```

```
cout<<" deletion begin"<<endl;
deletion(head,6);
deletion(head,8);
deletion(head,1);
deletion(head,7);
```

```
deletion(head,3);
display(head);
```

OUTPUT