```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
struct node *prev;
};
struct node *head=NULL;
void insertleft()
{
struct node *new_node;
new_node=(struct node*)malloc(sizeof(struct node));
printf("Enter the number: \n");
scanf("%d",&new_node->data);
new_node->next=NULL;
new_node->prev=NULL;

if(head==NULL)
{
head=new_node;
}
else
{
new_node->next=head;
head->prev=new_node;
head=new_node;
}

}
```

```
void del()
{
struct node *temp;
int elem;
if(head==NULL)
{
printf("Empty List \n");
return;
}
printf("Enter the element to be deleted\n");
scanf("%d",&elem);
temp=head;
while(temp->data!=elem)
{
temp=temp->next;
if(temp==NULL)
{
printf("Element is not in the list\n");
return;
}
}
if(temp==head)
{
head=head->next;
}
else if(temp->next==NULL)
{
temp=temp->prev;
temp->next=NULL;
}

else
{
temp->prev->next=temp->next;
temp->next->prev=temp->prev;
```

```
}
}
void insert_betweenL()
{
int listele;
struct node *new_node,*temp;
printf("Enter the element in the list\n");
scanf("%d",&listele);
new_node=(struct node*)malloc(sizeof(struct
node));
printf("Enter the new node data\n");
scanf("%d",&new_node->data);
new_node->next=NULL;
new_node->prev=NULL;
if(head==NULL)
{
printf("Empty list\n"); return;
}
temp=head;
while(temp->data!=listele)
{
temp=temp->next;
if(temp==NULL)
{
printf("Element is  not in the list");
return;
}
}
new_node->prev=temp->prev;
temp->prev=new_node;
new_node->next=temp;
new_node->prev->next=new_node;
}

void insert_betweenR()
{
```

```
int listele;
struct node *new_node,*temp;
printf("Enter the element in the list\n");
scanf("%d",&listele);
new_node=(struct node*)malloc(sizeof(struct
node));
printf("Enter the new node data\n");
scanf("%d",&new_node->data);
new_node->next=NULL;
new_node->prev=NULL;
if(head==NULL)
{
printf("Empty list\n"); return;
}
temp=head;
while(temp->data!=listele)
{
temp=temp->next;
if(temp==NULL)
{
printf("Element is  not in the list");
return;
}
}
new_node->next=temp->next;
temp->next=new_node;
new_node->prev=temp;
new_node->next->prev=new_node;
}
void display()
{
struct node *temp;
temp=head;
while(temp!=NULL)
{
printf("%d\t",temp->data);
```

```
temp=temp->next;
}
printf("\n");
}


int main()
{
int choice;
while(1)
{

printf(" 1. Insert to the left in the beginning \n");
printf(" 2.insert a node before a given node\n");
printf(" 3.insert a node after a given node\n");
printf(" 4. delete \n");
printf(" 5. display\n");
printf(" 6. exit\n");
printf("\nEnter your choice: \n");
scanf("%d",&choice);
switch(choice)
{
case 1: insertleft(); break;
case 2:insert_betweenL();break;
case 3:insert_betweenR();break;
case 4: del(); break;
case 5: display(); break;
case 6: exit(0);
}
}
}
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
1
Enter the number:
44
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
5
44      33      22
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
2
Enter the element in the list
33
Enter the new node data
5
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
5
44      5       33      22
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
3
Enter the element in the list
33
Enter the new node data
6
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
5
44      5       33      6       22
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
1
Enter the number:
22
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
1
Enter the number:
33
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
4
Enter the element to be deleted
22
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
5
44      5       33      6
```

```
 1. Insert to the left in the beginning
 2.insert a node before a given node
 3.insert a node after a given node
 4. delete
 5. display
 6. exit

Enter your choice:
6


...Program finished with exit code 0
Press ENTER to exit console.
```