```
#include<stdio.h>
#include<stdlib.h>

typedef struct nd {
        int data;
        struct nd *prev;
        struct nd *next;
} node ; // definition of a node

node *start; // global declaration, so that any function can access this

void display() // displaying the list
{
        node *p;
        if(start==NULL)
        {
                printf("List is empty!");
                exit(0);
        }
        p = start;
        while(p!=NULL)
        {
                printf("%d\n",p->data);
                p = p->next;
        }
}

void addatend() // inserting an element at the end position
{
        node *p,*temp;
        int val;
        printf("Enter the value..");
        scanf("%d",&val);
        temp = (node*)malloc(sizeof(node)); // creation of a node
        temp->data = val;
        p = start;
        while(p->next!=NULL)
                p = p->next;
        p->next = temp;
        temp->next = NULL;
        temp->prev = p;
}

void addatbeg() // inserting an element at the begining
{
        node *p,*temp;
        int val;
        printf("Enter the value..");
        scanf("%d",&val);
        temp = (node*)malloc(sizeof(node)); // creation of a node
        temp->data = val;
        temp->next = start;
        temp->prev = NULL;
        start->prev = temp;
        start = temp;
}

void addatpos() // inserting an element at a given position
{
        int i;
        node *p,*temp;
        int val,pos;
        printf("Enter the value..");
        scanf("%d",&val);
        printf("Enter the position..");
        scanf("%d",&pos);
        temp = (node*)malloc(sizeof(node)); // creation of a node
        temp->data = val;
        p = start;
        for(i=1;i<pos-1 && p!=NULL; i++)
                p = p->next;
        if(p==NULL)
                printf("There are less than %d elements!\n",pos);
        else
        {
                temp->next = p->next;
                p->next = temp;
                temp->next->prev = temp;
                temp->prev = p;
        }
}

void delatbeg() // deleting the 1st node
{
        node *p;
        if(start==NULL)
        {
                printf("List is empty!");
                exit(1);
        }
        p = start;
        start = start->next;
        start->prev = NULL;
        free(p);
}

void delatend() // deleting the last node
{
        node *p,*q;
        if(start==NULL)
        {
                printf("List is empty!");
```

```
            exit(2);
        }
        p = start;
        while(p->next!=NULL)
        {
            q = p; // q holds the previous node
            p = p->next;
        }
        q->next = NULL;
        free(p);
}

void delatpos() // deleting node from a given position
{
        node *p,*q;
        int i,pos;
        printf("Enter the position..");
        scanf("%d",&pos);
        if(start==NULL)
        {
            printf("List is empty!");
            exit(3);
        }
        p = start;
        for(i=1;i<pos && p!=NULL; i++)
        {
            q = p; // q holds the previous node
            p = p->next;
        }
        if(p==NULL)
            printf("There are less than %d elements!\n",pos);
        else
        {
            q->next = p->next;
            p->next->prev = q;
            free(p);
        }
}

void main()
{
        int val;

        start = NULL;
        start = (node*)malloc(sizeof(node)); // creation of the 1st node
        printf("Enter item..");
        scanf("%d",&val);
        start->data = val;
        start->prev = NULL;
        start->next = NULL;
        printf("Now add a node at the begining..\n");
        addatbeg();
        printf("The list is..\n");
        display();

        printf("Now add a node at the end..\n");
        addatend();
        printf("New list is..\n");
        display();

        printf("Now add a node at any position of list..\n");
        addatpos();
        printf("New list is..\n");
        display();

        printf("Deleting the first node..\n");
        delatbeg();
        printf("New list is..\n");
        display();

        printf("Deleting the last node..\n");
        delatend();
        printf("New list is..\n");
        display();

        printf("Deleting any node..\n");
        delatpos();
        printf("New list is..\n");
        display();

}
```