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

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

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

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

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

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

void addatpos() // inserting an element at a given position
{
        int i,val,pos;
        node *p,*temp;
        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 = last->link;
        for(i=1;i<=pos-2; i++)
                p = p->link;
        temp->link = p->link;
        p->link = temp;
}

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

void delatend() // deleting the last node
{
        node *p,*q;
        if(last==NULL)
        {
                printf("List is empty!");
                exit(2);
        }
        p = last->link;
        do
        {
                q = p; // q holds the previous node
                p = p->link;
        } while(p->link!=last->link);
        q->link = last->link;
        last=q;
        free(p);
}

void delatpos() // deleting node from a given position
{
        node *p,*q;
        int i,pos;
        printf("Enter the position..");
        scanf("%d",&pos);
        if(last==NULL)
        {
                printf("List is empty!");
                exit(3);
        }
        p = last->link;
```

```
        for(i=1; i<pos; i++)
        {
                q = p; // q holds the previous node
                p = p->link;
        }
        q->link = p->link;
        free(p);
}

void main()
{
        int val;

        last = NULL;
        last = (node*)malloc(sizeof(node)); // creation of the 1st node
        printf("Enter the value..");
        scanf("%d",&val);
        last->data = val;
        last->link = last;

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

        printf("Now add a node at the begining..\n");
        addatbeg();
        printf("The 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();

}
```