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

struct node
{
    struct node *prev;
    int data;
    struct node *next;
};
struct node *head=NULL;
void insertbegindoubly();
void displayforward();
void insertposition(int,int,struct node *);
void insertend(int);
void displaybackward();
void deletedobly();

int main()
{
     int item,n;
    insertbegindoubly();
    insertbegindoubly();
    insertbegindoubly();
    insertbegindoubly();
    displayforward();
    displaybackward();
    insertposition(n,item,head);
    displayforward();
    printf("\nin backwards\n");
    displaybackward();
    insertend(item);
    displayforward();



    return 0;
}

void insertbegindoubly()
{
    struct node *newnode;
    newnode=(struct node *)malloc(sizeof(struct node));
    printf("enter data : ");
    scanf("%d",&newnode->data);
    newnode->next=NULL;
    if(head==NULL){
        head=newnode;
    }

    else{
        newnode->next=head;
        head->prev=newnode;
        head=newnode;
    }
}

void displayforward()
{
    struct node *temp;
    if(head==NULL){
        printf("List is empty\n");
    }
    else{
        temp=head;
        printf("list elements in doubly linked list are:\n");
        while(temp!=NULL){
```

```
            printf("%d\t",temp->data);
            temp=temp->next;
        }
    }
}

void insertposition(int pos,int data,struct node *head)
{
    struct node *temp;
    struct node *newnode=(struct node *)malloc(sizeof(struct node));
    newnode->next=NULL;
    printf("\nenter at position:");

    scanf("%d",&pos);
    if(pos==1){
            insertbegindoubly();

    }
    else{
             pos=pos-1;
    printf("enter data:");
    scanf("%d",&newnode->data);
    newnode->next=NULL;
    if(pos==0)
        {
            head=newnode;
        }
    else{
        temp=head;
        while(--pos)
            {
                temp=temp->next;
            }
        newnode->next = temp->next;
        temp->next = newnode;
        }

    }

}

void insertend(int item)
{
    struct node *temp;
    struct node *newnode=(struct node *)malloc(sizeof(struct node *));
    printf("\nenter data for end node ");
    scanf("%d",&newnode->data);
    if(head==NULL){
            newnode->next=NULL;
            head=newnode;
    }
    else{
            temp=head;
            while(temp->next!=NULL){
            temp=temp->next;
            }
    temp->next=newnode;
    newnode->next=NULL;
    }
}

void displaybackward()
{
    struct node *temp1=head;

        while(temp1->next!=NULL){
```

```
            temp1=temp1->next;
        }
            while(temp1!=head){
                printf("%d\t",temp1->data);
                temp1=temp1->prev;
            }
            printf("%d\t",temp1->data);



}
void deletedobly();
{
    struct node*temp;
    if(head==NULL)
    {
        printf("list is empity")
    }
    else if(head->next==NULL)
    {
        head=NULL;
        free(head);
        printf("\n begin node deleted");
    }
    else
    {
        temp=head;
        head=head->next;
        head->prev=NULL;
        free(temp);
        print("\n begin node deleted");
    }
}

```