**Program:**

#include<stdio.h>

#include<stdlib.h>

struct Node {

    int data;

    struct Node \* next;

    struct Node \* prev;

};

void display(struct Node \* head){

    struct Node \* p = head;

    while (p != NULL) {

        printf("%d ", p->data);

        p=p->next;

    }

    printf("\n \n");

}

struct Node \* insertAtStart(struct Node \* head){

    int data;

    printf("Enter data to be inserted at Start ");

    scanf("%d", &data);

    struct Node \* newnode = malloc(sizeof(struct Node));

    newnode->data=data;

    newnode->prev=NULL;

    newnode->next=head;

    if(head != NULL) {

        head->prev=newnode;

    }

    head=newnode;

    printf("%d is inserted at Start\n \n",data);

    return head;

}

struct Node \* insertAtEnd(struct Node \* head){

    int data;

    printf("Enter data to be inserted at End ");

    scanf("%d", &data);

    struct Node \* newnode = malloc(sizeof(struct Node));

    newnode->data=data;

    newnode->next=NULL;

    if (head==NULL)

    {

        newnode->prev=NULL;

        head=newnode;

    }

    else{

        struct Node \* p = head;

        while(p->next !=NULL){

            p=p->next;

        }

        newnode->prev=p;

        p->next=newnode;

    }

    printf("%d is inserted at End\n \n",data);

    return head;

}

struct Node \* insertAtPosition(struct Node \* head){

    int data, position;

    printf("Enter data to be inserted ");

    scanf("%d", &data);

    printf("Enter the position you want to insert data ");

    scanf("%d",&position);

    struct Node \* newnode = malloc(sizeof(struct Node));

    newnode->data=data;

    struct Node \* p = head;

    int i;

    for ( i = 1; i < position; i++)

    {

        p=p->next;

    }

    newnode->prev=p;

    newnode->next=p->next;

    (p->next)->prev=newnode;

    p->next=newnode;

    printf("%d is inserted at %d \n \n",data, position);

    return head;

}

struct Node \* deleteAtStart(struct Node \* head){

    if (head==NULL)

    {

        printf("No elements present to delete\n \n");

    }

    else{

        printf("%d is deleted from Start \n \n", head->data);

        head=head->next;

    }

    return head;

}

struct Node \* deleteAtEnd(struct Node \* head){

    if (head==NULL)

    {

        printf("No elements present to delete\n \n");

    }

    else{

        struct Node \* p = head;

        while ((p->next)->next != NULL){

            p=p->next;

        }

        printf("%d is deleted from End \n \n", (p->next)->data);

        p->next=NULL;

    }

    return head;

}

struct Node \* deleteAtPosition(struct Node \* head){

    if (head==NULL)

    {

        printf("No elements present to delete\n \n");

    }

    else{

        int position;

        printf("Enter the position you want to delete data ");

        scanf("%d",&position);

        struct Node \* p = head;

        int i;

        for ( i = 1; i < position && p != NULL; i++)

        {

            p=p->next;

        }

        if (p == NULL) {

            printf("Position exceeds length of linked list\n \n");

            return head;

        }

        printf("%d is deleted at %d \n \n", (p->next)->data, position);

        ((p->next)->next)->prev=p;

        p->next=(p->next)->next;

    }

    return head;

}

int  main()

{

    struct Node \* head = NULL;

    int choice=-1;

    while (choice != 10)

    {

        printf("Enter 0 to display \nEnter 1 to insert at start \nEnter 2 to insert at End \nEnter 3 to insert at any position \nEnter 4 to delete at Start \nEnter 5 to delete at End \nEnter 6 to delete any position \nEnter 10 to exit \n");

        scanf("%d", &choice);

        if (choice==0){

            display(head);

        }

        else if (choice==1)

        {

            head=insertAtStart(head);

        }else if (choice==2)

        {

            head=insertAtEnd(head);

        }

        else if (choice==3)

        {

            head=insertAtPosition(head);

        }

        else if (choice==4)

        {

            head=deleteAtStart(head);

        }

        else if (choice ==5)

        {

            head=deleteAtEnd(head);

        }

        else if (choice==6)

        {

            head=deleteAtPosition(head);

        }

        else if(choice==10){

            break;

        }

    }

    return 0;

}

**Output screenshots:**

 



 

