LAB PROGRAM 7 a

Q) WAP to implement doubly linked list with primitive operations

1.Create a doubly linked list

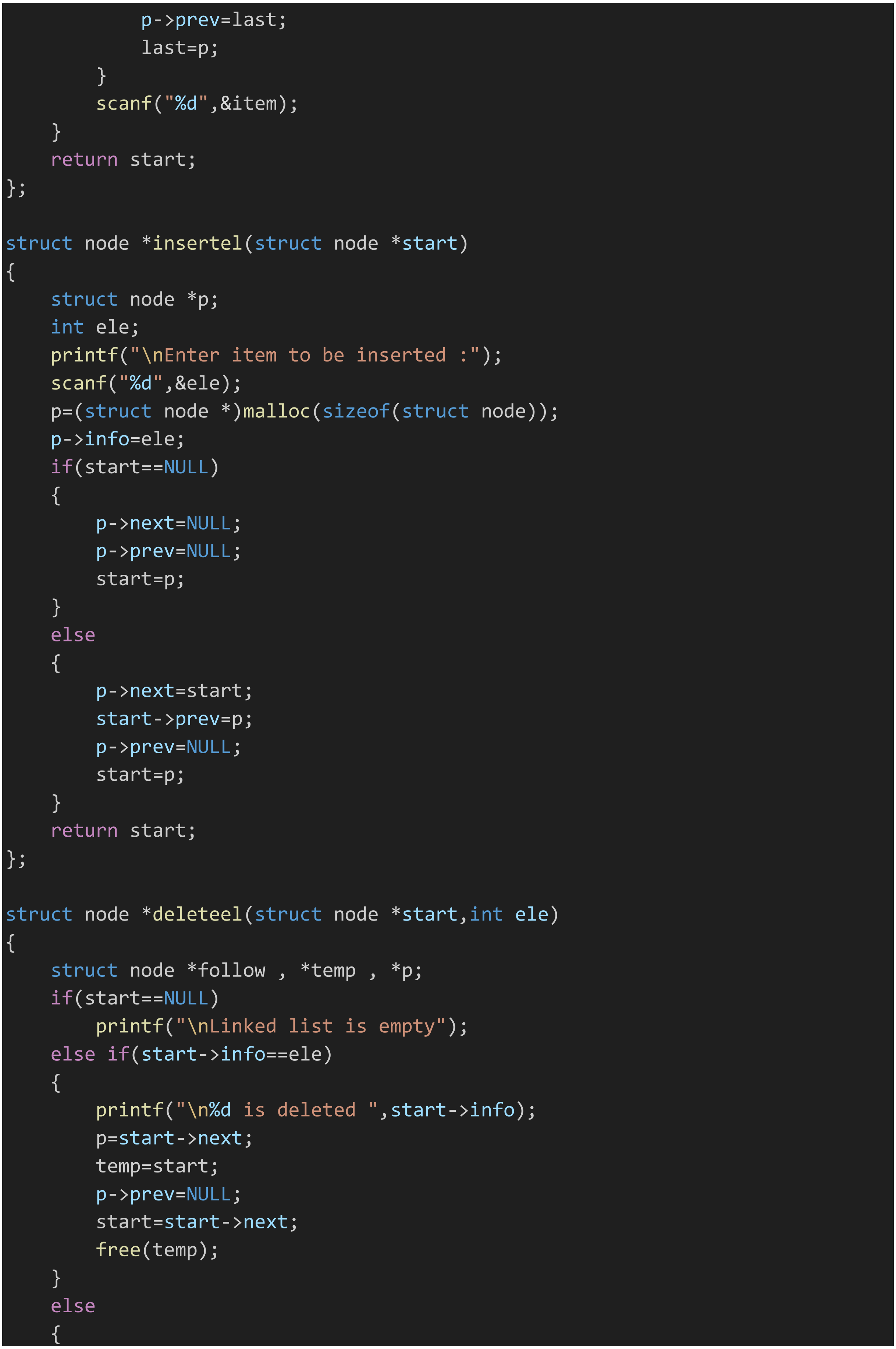
2.Insert node to left of new node

3.Delete node based on value

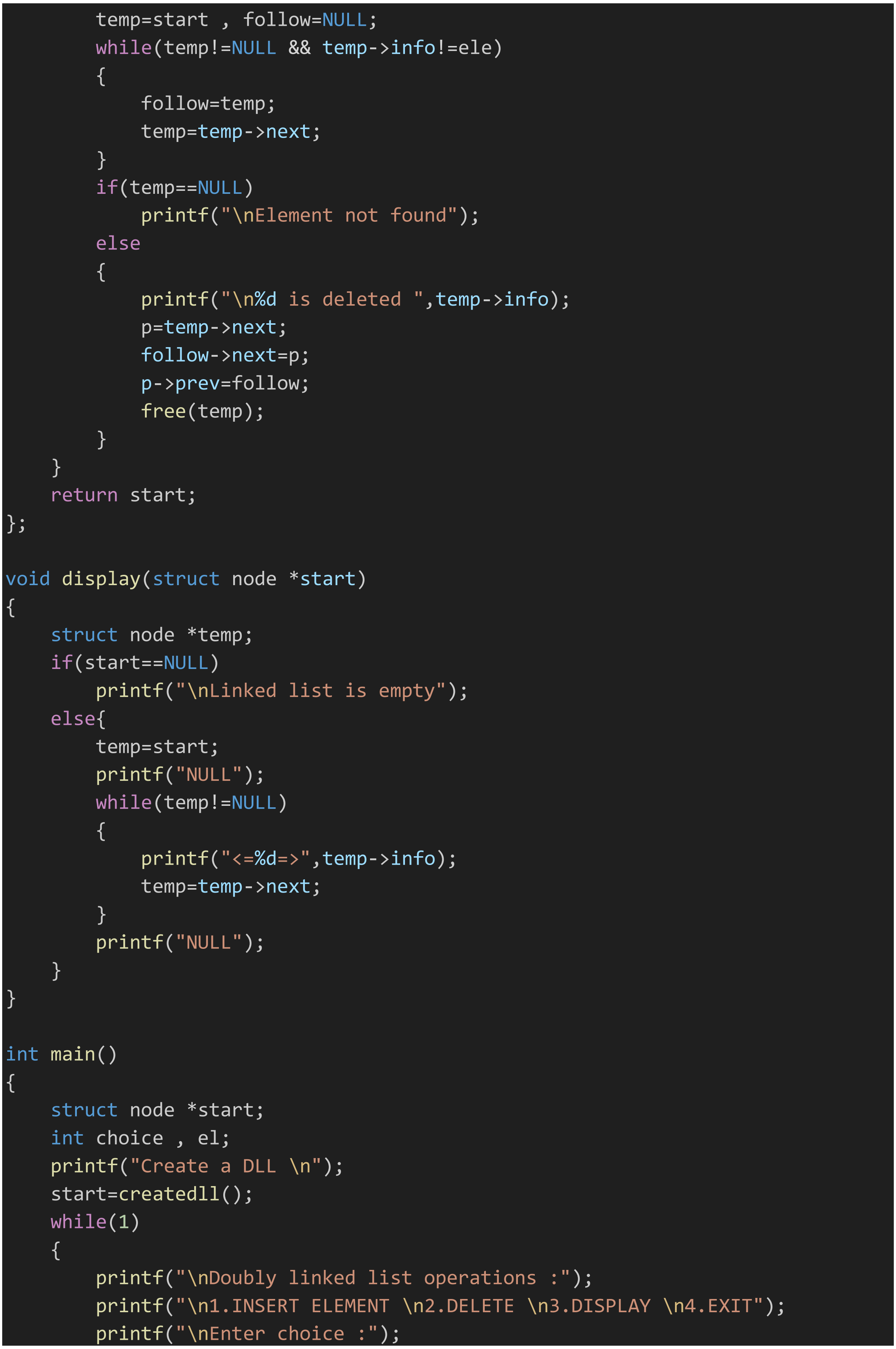
4.Display content

CODE

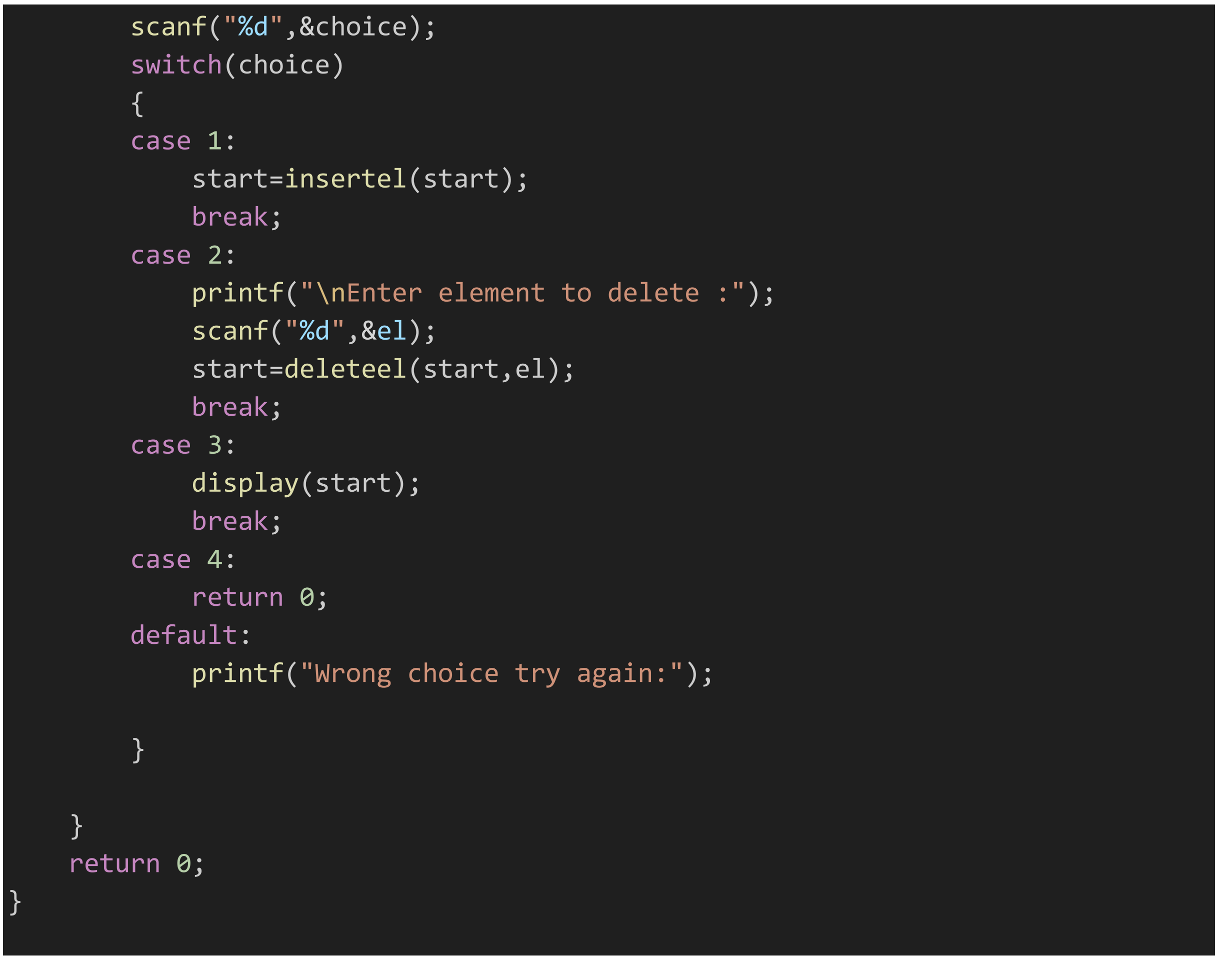
```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int info;
    struct node *prev;
    struct node *next;

};

struct node * createdll()
{
    struct node *start=NULL , *last ,*p;
    int item;
    printf("\nEnter item to be inserted to the doubly linked list (-999 to exit) :");
    scanf("%d",&item);
    while(item!=-999)
    {
        p=(struct node *)malloc(sizeof(struct node));
        p->info=item;
        if(start==NULL)
        {
            p->next=NULL;
            p->prev=NULL;
            start=p;
            last=p;
        }
        else
        {
            p->next=NULL;
            last->next=p;
```



```
            p->prev=last;
            last=p;
        }
        scanf("%d",&item);
    }
    return start;
};

struct node *insertel(struct node *start)
{
    struct node *p;
    int ele;
    printf("\nEnter item to be inserted :");
    scanf("%d",&ele);
    p=(struct node *)malloc(sizeof(struct node));
    p->info=ele;
    if(start==NULL)
    {
        p->next=NULL;
        p->prev=NULL;
        start=p;
    }
    else
    {
        p->next=start;
        start->prev=p;
        p->prev=NULL;
        start=p;
    }
    return start;
};

struct node *deleteel(struct node *start,int ele)
{
    struct node *follow , *temp , *p;
    if(start==NULL)
        printf("\nLinked list is empty");
    else if(start->info==ele)
    {
        printf("\n%d is deleted ",start->info);
        p=start->next;
        temp=start;
        p->prev=NULL;
        start=start->next;
        free(temp);
    }
    else
    {
```

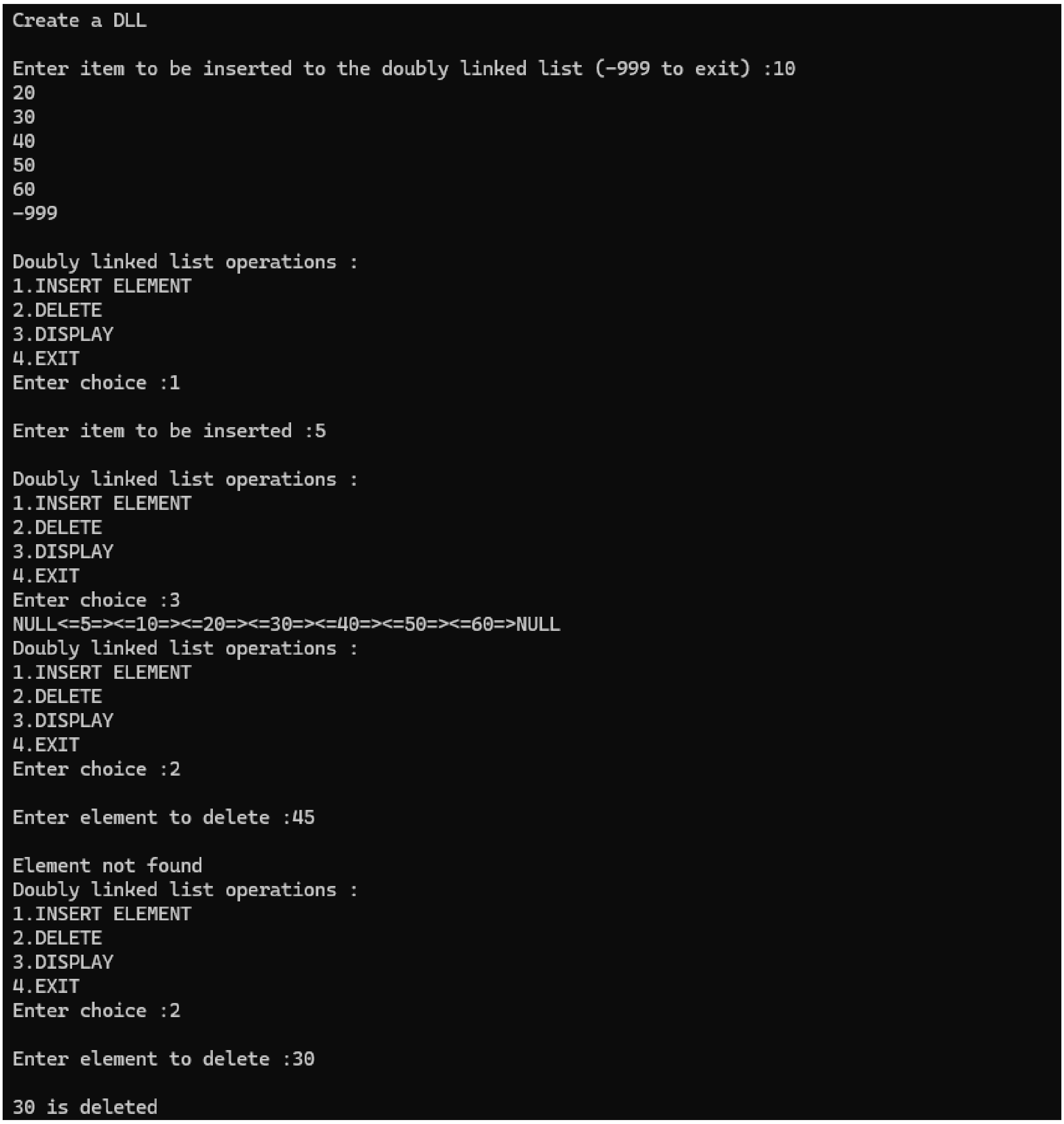


```
        temp=start , follow=NULL;
        while(temp!=NULL && temp->info!=ele)
        {
            follow=temp;
            temp=temp->next;
        }
        if(temp==NULL)
            printf("\nElement not found");
        else
        {
            printf("\n%d is deleted ",temp->info);
            p=temp->next;
            follow->next=p;
            p->prev=follow;
            free(temp);
        }
    }
    return start;
};

void display(struct node *start)
{
    struct node *temp;
    if(start==NULL)
        printf("\nLinked list is empty");
    else{
        temp=start;
        printf("NULL");
        while(temp!=NULL)
        {
            printf("<=%d=>",temp->info);
            temp=temp->next;
        }
        printf("NULL");
    }
}

int main()
{
    struct node *start;
    int choice , el;
    printf("Create a DLL \n");
    start=createdll();
    while(1)
    {
        printf("\nDoubly linked list operations :");
        printf("\n1.INSERT ELEMENT \n2.DELETE \n3.DISPLAY \n4.EXIT");
        printf("\nEnter choice :");
```

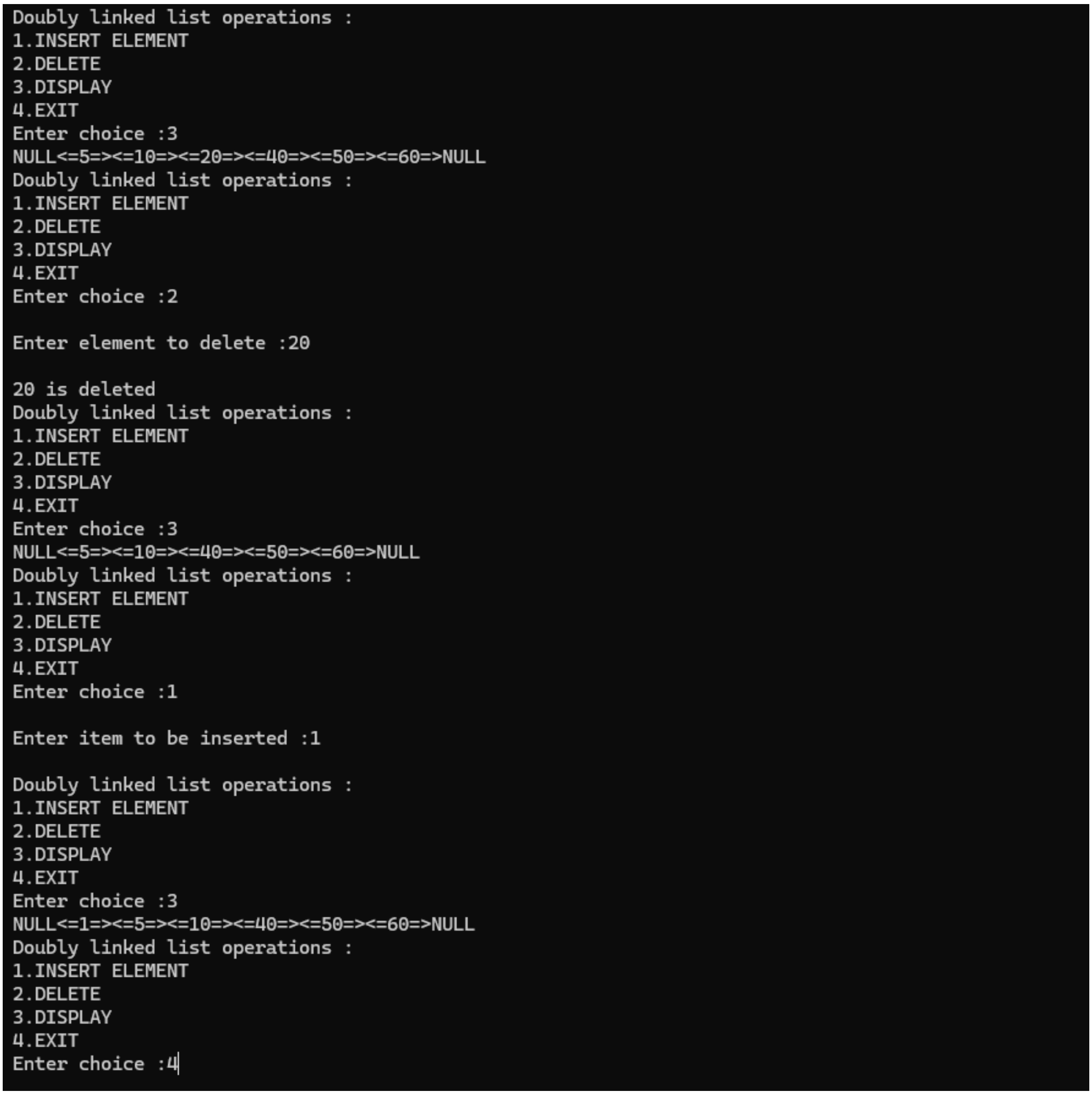


```
        scanf("%d",&choice);
        switch(choice)
        {
        case 1:
            start=insertel(start);
            break;
        case 2:
            printf("\nEnter element to delete :");
            scanf("%d",&el);
            start=deleteel(start,el);
            break;
        case 3:
            display(start);
            break;
        case 4:
            return 0;
        default:
            printf("Wrong choice try again:");

        }

    }
    return 0;
}
```

OUTPUT



```
Create a DLL

Enter item to be inserted to the doubly linked list (-999 to exit) :10
20
30
40
50
60
-999

Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :1

Enter item to be inserted :5

Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :3
NULL<=5=><=10=><=20=><=30=><=40=><=50=><=60=>NULL
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :2

Enter element to delete :45

Element not found
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :2

Enter element to delete :30

30 is deleted
```



```
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :3
NULL<=5=><=10=><=20=><=40=><=50=><=60=>NULL
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :2

Enter element to delete :20

20 is deleted
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :3
NULL<=5=><=10=><=40=><=50=><=60=>NULL
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :1

Enter item to be inserted :1

Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :3
NULL<=1=><=5=><=10=><=40=><=50=><=60=>NULL
Doubly linked list operations :
1.INSERT ELEMENT
2.DELETE
3.DISPLAY
4.EXIT
Enter choice :4
```