

```
/*doubly link list copy*/
```

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

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

```
struct node
```

```
{
```

```
    int data;
```

```
    struct node*link;
```

```
};
```

```
struct node*header;
```

```
struct node*header1;
```

```
struct node*create_ll(struct node*);
```

```
struct node*display(struct node*);
```

```
struct node*copy(struct node*,struct node*);
```

```
int main()
```

```
{
```

```
    int choice=0;
```

```
    while(choice!=4)
```

```
    {
```

```
        printf("**main menu**\n");
```

```
        printf("1.create list\n2.display the list\n3.copy the linked list into another linked  
list\n4.exit\n");
```

```
        printf("enter your choice\n");
```

```
        scanf("%d",&choice);
```

```
        switch(choice)
```

```
        {
```

```
            case 1:header=create_ll(header);
```

```
            break;
```

```
            case 2:header=display(header);
```

```
            break;
```

```
            case 3:header1=copy(header1,header);
```

```
            break;
```

```

        case 4:exit(0);

        default:

            printf("invalid choice\n");

    }

}

struct node*create_ll(struct node*header)
{

    struct node*new_node,*ptr;

    int item;

    printf("enter -1 to end\n");

    printf("enter the data: \n");

    scanf("%d",&item);

    while(item!=-1)

    {

        new_node=(struct node*)malloc(sizeof(struct node*));

        new_node->data=item;

        if(header==NULL)    //list is empty

        {

            new_node->link=NULL;

            header=new_node;

        }

        else

        {

            ptr=header;

            while(ptr->link!=NULL)

            {

                ptr=ptr->link;

            }

            ptr->link=new_node;

            new_node->link=NULL;

```

```

        }

        printf("enter the data: \n");

        scanf("%d",&item);

    }

    printf("link list is created\n");

    return header;
}

struct node*display(struct node*header)
{

    printf("the linked list is below\n");

    struct node*ptr;

    ptr=header;

    while(ptr!=NULL)    //list is not empty
    {

        printf("%d\n",ptr->data);

        ptr=ptr->link;

    }

    return header;
}

struct node*copy(struct node*header,struct node*header1)
{

    struct node*new_node;

    struct node*ptr,*ptr1;

    new_node=(struct node*)malloc(sizeof(struct node*));

    new_node->data=NULL;

    header1=new_node;

    ptr1=header1;

    ptr=header;

    while(ptr!=NULL)
    {

        header1->data=ptr->data;

```

```

        ptr1->link=header1;

        ptr1=header1;

        ptr=ptr->link;

    }

    printf("list is copied\n");

    return header1;

}

```

```

C:\Users\HP\OneDrive\Desktop\collage work 3rd sem\doubly linked list copy new.exe
**main menu**
1.create list
2.display the list
3.copy the linked list into another linked list
4.exit
enter your choice
1
enter -1 to end
enter the data:
10
enter the data:
20
enter the data:
30
enter the data:
40
enter the data:
50
enter the data:
60
enter the data:
-1
link list is created
**main menu**
1.create list
2.display the list
3.copy the linked list into another linked list
4.exit
enter your choice
2
the linked list is below
10
20
30
40
50
60
**main menu**
1.create list
2.display the list
3.copy the linked list into another linked list
4.exit
enter your choice
3
list is copied
**main menu**
1.create list
2.display the list
3.copy the linked list into another linked list
4.exit
enter your choice
2
the linked list is below
10
20
30
40
50
60
**main menu**
1.create list
2.display the list
3.copy the linked list into another linked list
4.exit
enter your choice
4

-----
Process exited after 23.87 seconds with return value 0
Press any key to continue . . .

```