Double Linked List

```
#include<stdio.h>
#include<malloc.h>
struct node
{
    struct node *prev;
    int data;
    struct node *next;
};
struct node *start;
void insertbeg(void)        // insert element at the beginning of DLL
{
    int a;
    struct node *nn;
    nn=(struct node *)malloc(sizeof(struct node));
    printf("enter data:");
    scanf("%d",&nn->data);
    a=nn->data;
    if(start==NULL)          //If Linked list is empty
    {
        nn->prev=nn->next=NULL;
        start=nn;
    }
```

```
    else
    {
        nn->next=start;
        nn->prev=NULL;
        start->prev=nn;
        start=nn;
    }
    printf("%d succ. inserted\n",a);
    return;
}
void insertend(void)        // insert element at the End of DLL
{
    int b;
    struct node *nn,*lp;
    nn=(struct node *)malloc(sizeof(struct node));
    printf("enter data:");
    scanf("%d",&nn->data);
    b=nn->data;
    if(start==NULL)
    {
        nn->prev=nn->next=NULL;
        start=nn;
    }
    else
```

```
    {
        lp=start;
        while(lp->next!=NULL)
        {
                lp=lp->next;
        }
        nn->prev=lp;
        lp->next=nn;
        nn->next=NULL;
    }
    printf("%d succ. inserted\n",b);
    return;
}
void insertmid(void)// insert element before the given element of
DLL
{
    struct node *nn,*temp,*ptemp;int x,c;
    if(start==NULL)
    {
        printf("dll is empty\n"); return;
    }
    printf("enter data before which new node is to be insreted\n");
    scanf("%d",&x);
    if(x==start->data)
```

```
{
    insertbeg();
    return;
}
ptemp=start;
temp=start->next;
while(temp!=NULL&&temp->data!=x)
{
    ptemp=temp;
    temp=temp->next;
}
if(temp==NULL)
{
    printf("%d data does not exist\n",x);
}
else
{
    nn=(struct node *)malloc(sizeof(struct node));
    printf("enter data");
    scanf("%d",&nn->data);
    c=nn->data;
    nn-> prev=ptemp;
    nn->next=temp;
    ptemp->next=nn;
```

```
        temp->prev=nn;
        printf("%d succ. inserted\n",c);
    }
    return;
}
void deletion(void)
{
    struct node *pt,*t,*nt;
    int x;
    if(start==NULL)
    {
        printf("dll is empty\n");
        return;
    }
    printf("enter data to be deleted");
    scanf("%d",&x);
    if(x==start->data)
    {
        t=start;
        start=start->next;
        free(t);
        printf("%d is succ. deleted\n",x);
        if(start!=NULL)
        {
```

```
        start->prev=NULL;
    }
    return;
}
pt=start;
t=start->next;
while(t!=NULL&&t->data!=x)
{
    pt=t;
    t=t->next;
}
if(t==NULL)
{
    printf("%d does not exist\n",x);return;
}
else
{
    pt->next=t->next;
    if(t->next!=NULL)
    {
        nt=t->next;
        nt->prev=pt;
    }
    free(t);
```

```
    }
    printf("%d is succ. deleted\n",x);
    return;
}
void display(void)
{
    struct node *temp;
    if(start==NULL)
    {
        printf("dll is empty\n");
        return;
    }
    printf("displaying in forword order\n");
    temp=start;
    while(temp!=NULL)
    {
                    printf("%d\t",temp->data);
                    temp=temp->next;
    }
    printf("\n\n");
    return;
}
int  main()
{
```

```
int c,a;
 start=NULL;
do
{

    printf("\n1:insert\n2:delete\n3:display\n4:exit\nenter choice:\n\n");
                scanf("%d",&c);
                switch(c)
                {
                    case 1:
printf("\t\t1:insertbeg\t\t\n2:insert end\n\t\t3:insert mid\nenter choice:\n\n");
                            scanf("%d",&a);
                    switch(a)
                    {
                            case 1:insertbeg();break;
                            case 2:insertend();break;
                            case 3:insertmid();break;
                    }
                    break;
                    case 2:deletion();break;
                   case 3:display();break;
                   case 4:printf("program ends\n");break;
                   default:printf("wrong choice\n");
```

```
            break;
        }
    }while(c!=4) ;
return 0;
}
```