Singly linked list

```
# include<stdio.h>
# include<stdlib.h>
struct node
{
    int data;
    struct node *nt;
}*st, *nn, *tp;
int main ()
{
    int i,ch;
    do
    {
nn = (struct node *)malloc(sizeof(struct node));
printf("Enter the data of new node: ");
scanf("%d", &nn->data);
if (st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->nt=nn;
tp=tp->nt;
}
printf("Enter any non-zero value to add another node. Enter 0 otherwise: ");
scanf("%d",&ch);
}while(ch!=0);
tp->nt=NULL;
printf("The data elements of the linked list are: ");
tp=st;
while (tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->nt;
}
return 0;
}
```

Output:

```
Enter the data of new node: 7
Enter any non-zero value to add another node. Enter 0 otherwise: 7
Enter the data of new node: 77
Enter any non-zero value to add another node. Enter 0 otherwise: 7
Enter the data of new node: 777
Enter any non-zero value to add another node. Enter 0 otherwise: 7
Enter the data of new node: 7777
Enter any non-zero value to add another node. Enter 0 otherwise: 0
The data elements of the linked list are: 7     77     777     7777
```

Reverse a Singly linked list

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
}*st,*tp,*nn;
void revers()
{
   nn=st;
   tp=st->next;
   st=tp->next;
   tp->next=nn;
   nn->next=NULL;
   while(st!=NULL)
   {
      nn=tp;
      tp=st;
```

```
        st=st->next;
        tp->next=nn;
    }
    st=tp;
    printf("The reversed list is: ");
    while(tp!=0)
    {
        printf("%d\t",tp->data);
        tp=tp->next;
    }
    }
int main()
{
    int ch;
        do
    {
        nn=(struct node *)malloc(sizeof(struct node));
    printf("Enter the element:");
    scanf("%d",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
    }
    else
    {
    tp->next=nn;
    tp=tp->next;
    }
        printf("Enter 0 if no more elements. Enter any value otherwise:");
        scanf("%d",&ch);
    }while(ch!=0);
tp->nt=NULL;

        printf("\nThe linked list elements are: ");
        tp=st;
    while(tp!=0)
    {
    printf("%d\t",tp->data);
```

```
   tp=tp->next;
   }
   revers();
     return 0;
}
```

Output:

```
Enter the element:7

Enter 0 if no more elements. Enter any value otherwise:7

Enter the element:77

Enter 0 if no more elements. Enter any value otherwise:7

Enter the element:777

Enter 0 if no more elements. Enter any value otherwise:7

Enter the element:7777

Enter 0 if no more elements. Enter any value otherwise:0

The linked list elements are: 7      77       777      7777     The reversed list is: 7777   777      77       7
```

Insert_at_beg_SLL

```
#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node *next;

}*st, *tp, *nn;

void create()

{

int ch;

do

{

nn=malloc(sizeof(struct node));

printf("Enter a data element: ");

scanf("%d",&nn->data);

if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->next=nn;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void display()
{
tp=st;
printf("The linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
void insert_at_beg()
{
   nn=malloc(sizeof(struct node));
   printf("Enter the node you wish to insert at beginning");
   scanf("%d",&nn->data);
   nn->next=st;
   st=nn;
   }
int main()
```

```
{
create();
insert_at_beg();
display();
return 0;
}
```

Insert_at_end_SLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*st, *tp, *nn;
void create()
{
int ch;
do{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->next=nn;
tp=tp->next;
//tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void insert_at_end()
{
    nn=malloc(sizeof(struct node));
    printf("Enter the node you wish to insert at end: ");
    scanf("%d",&nn->data);
    tp->next=nn;
    tp=tp->next;
    tp->next=NULL;
}
void display()
{
tp=st;
printf("The linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_at_end();
display();
return 0;
}
```

Insert_at_before_an_element_SLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
```

```
{
int data;
struct node *next;
}*st, *tp, *nn, *rn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->next=nn;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void insert_before_element()
{
   int ref;
   printf("Enter the reference value");
   scanf("%d",&ref);
   tp=st;
```

```
    while(tp!=NULL)
    {
        if(tp->data==ref)
        rn=tp;
        tp=tp->next;
    }
       if (rn==NULL)
        printf("Entered element is not present in the linked list. Insertion not possible");
    else
    {
    nn=malloc(sizeof(struct node));
    printf("Enter the data to insert before reference: ");
    scanf("%d",&nn->data);
    tp=st;
    while(tp!=NULL)
    {
        if(tp->next==rn)
        {
        tp->next=nn;
        nn->next=rn;
                }
    tp=tp->next;
    }
}
}
void display()
{
tp=st;
printf("The linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_before_element();
display();
return 0;
}
```

//Count the number of nodes in SLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*st, *tp, *nn, *rn;
void create()
{
int ch, count=0;;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
count++;
if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
```

```
tp->next=nn;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
printf("The number of nodes in SLL is: %d",count);
}
int main()
{
create();
return 0;
}
```

//Delete last node in SLL

```
# include<stdio.h>
# include<stdlib.h>
struct node
{
   int data;
   struct node *nt;
}*st, *nn, *tp;
void create()
{
    int ch;
   do
   {
nn = malloc(sizeof(struct node));
printf("Enter the data of new node: ");
scanf("%d", &nn->data);
if (st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->nt=nn;
tp=tp->nt;
}
printf("Enter any non-zero value to add another node. Enter 0 otherwise: ");
scanf("%d",&ch);
}while(ch!=0);
tp->nt=NULL;
}
void display()
{
   printf("The data elements of the linked list are: ");
   tp=st;
while (tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->nt;
}
}
void del_end()
{
   tp=st;
   while(tp->nt!=nn)
   tp=tp->nt;
   free(nn);
   tp->nt=NULL;
   nn=tp;
   printf("Last node is deleted\n");
}
int main ()
```

```
{
create();
del_end();
display();
return 0;
}
```

Doubly_linked_list_and_reverse

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void revers()
{
    tp=nn;
    printf("The reversed DLL is: ");
    while(tp!=NULL)
    {
        printf("%d\t",tp->data);
        tp=tp->prev;
    }
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}

int main()
{
create();
display();
revers();
```

```
return 0;
}
```

Doubly_linked_list_char

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
char data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%s",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void revers()
{
  tp=nn;
  printf("The reversed DLL is: ");
  while(tp!=NULL)
  {
    printf("%c\t",tp->data);
    tp=tp->prev;
  }
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%c\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
display();
revers();
return 0;
}
```

Doubly_linked_list_insert_at__beg

```
#include<stdio.h>
#include<stdlib.h>
struct node
```

```
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void revers()
{
   tp=nn;
   printf("The reversed DLL is: ");
```

```
    while(tp!=NULL)
    {
        printf("%d\t",tp->data);
        tp=tp->prev;
    }
}
void insert_at_beg()
{
    nn=malloc(sizeof(struct node));
    printf("Enter the node you wish to insert at beginning");
    scanf("%d",&nn->data);
    nn->next=st;
    st=nn;
    nn->prev=NULL;
    }
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_at_beg();
display();
return 0;
}
```

Doubly_linked_list_delete_a_node

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void deletee()
```

```
{
    struct node *deln;
    int delval;
   printf("Enter value to delete");
   scanf("%d",&delval);
    tp=st;
   while(tp!=NULL)
   {
     if(tp->data==delval)
   deln=tp;
   tp=tp->next;
   }
   tp=deln->prev;
   nn=deln->next;
   tp->next=nn;
   nn->prev=tp;
   free(deln);
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
deletee();
display();
return 0;
}
```

Doubly_linked_list_insert_at _end

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
void insert_at_end()
{
    nn=malloc(sizeof(struct node));
    printf("Enter the data to insert at end:");
    scanf("%d",&nn->data);
    tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
int main()
{
create();
insert_at_end();
display();
return 0;
}
```

insert_before_an_element_DLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
```

```
{

int data;

struct node *prev, *next;

}*st, *tp, *nn;

void create()

{

int ch;

do

{

nn=malloc(sizeof(struct node));

printf("Enter a data element: ");

scanf("%d",&nn->data);

if(st==NULL)

{

st=nn;

tp=nn;

tp->prev = NULL;

}

else

{

tp->next=nn;

nn->prev=tp;

tp=tp->next;

tp->next=NULL;

}

printf("Enter 0 to exit; any other value to continue: ");

scanf("%d",&ch);

}while(ch!=0);

}

void revers()

{

   tp=nn;

   printf("The reversed DLL is: ");
```

```
    while(tp!=NULL)

    {

        printf("%d\t",tp->data);

        tp=tp->prev;

    }

}

void insert_beforel()

{

    int sel;

    struct node *ins;

    printf("Enter the value before which you wish to insert:");

    scanf("%d",&sel);

    nn=malloc(sizeof(struct node));

    printf("Enter the node you wish to insert before %d:",sel);

    scanf("%d",&nn->data);

    tp=st;

    while(tp!=NULL)

    {

    if(tp->data==sel)

    ins=tp;

    tp=tp->next;

    }

tp=ins->prev;

tp->next=nn;

ins->prev=nn;

nn->prev=tp;

nn->next=ins;

    }

void display()

{

tp=st;

printf("The doubly linked list is: ");

while(tp!=NULL)
```

```
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_beforel();
display();
return 0;
}
```

Doubly linked list delete at end

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void del_end()
{
    tp=nn->prev;
    tp->next=NULL;
    free(nn);
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
del_end();
display();
```

```
return 0;
}
```

Doubly_linked_list_delete_a_node

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void deletee()
{
  int delval;
  printf("Enter value to delete");
  scanf("%d",&delval);
   tp=st;
  while(tp->data!=delval)
       tp=tp->next;
    nn=tp->next;
    tp->prev->next=nn;
    nn->prev=tp->prev;
  free(tp);
}
void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
deletee();
display();
return 0;
}
```

Output:

```
Enter a data element: 7

Enter 0 to exit; any other value to continue: 7

Enter a data element: 77

Enter 0 to exit; any other value to continue: 7

Enter a data element: 777

Enter 0 to exit; any other value to continue: 7777

Enter a data element: 7777

Enter 0 to exit; any other value to continue: 0

Enter value to delete77

The doubly linked list is: 7    777     7777

Process returned 0 (0x0)
```

insert_before_an_element_DLL_reduced

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void insert_beforel()
{
    int sel;
    printf("Enter the value before which you wish to insert:");
    scanf("%d",&sel);
    nn=malloc(sizeof(struct node));
    printf("Enter the node you wish to insert before %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while(tp->data!=sel)
    tp=tp->next;
    nn->prev=tp->prev;
    nn->next=tp;
    tp->prev->next=nn;
    tp->prev=nn;
}
   void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
```

```
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_beforel();
display();
return 0;
}
```

Output:

```
Enter a data element: 7
Enter 0 to exit; any other value to continue: 7
Enter a data element: 77
Enter 0 to exit; any other value to continue: 7
Enter a data element: 777
Enter 0 to exit; any other value to continue: 7
Enter a data element: 7777
Enter 0 to exit; any other value to continue: 0
Enter the value before which you wish to insert:77
Enter the node you wish to insert before 77:9
The doubly linked list is: 7    9       77      777     7777
Process returned 0 (0x0)
```

Circular_Doubly_linked_list_and_reverse

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
```

```
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void revers()
{
  int j,m;
  printf("Enter number of data elements to print in reverse:");
  scanf("%d",&m);
  tp=st;
```

```
   printf("The reversed DLL is: ");
   for(j=0;j<m;j++)
   {
       tp=tp->prev;
      printf("%d\t",tp->data);
        }
}
void display()
{
   int i,n;
   printf("Enter number of data elements to print:");
   scanf("%d",&n);
tp=st;
printf("The doubly linked list is: ");
for(i=0;i<n;i++)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
display();
revers();
return 0;
}
```

Output:

```
Enter a data element: 7
Enter 0 to exit; any other value to continue: 7
Enter a data element: 77
Enter 0 to exit; any other value to continue: 7
Enter a data element: 777
Enter 0 to exit; any other value to continue: 7
Enter a data element: 7777
Enter 0 to exit; any other value to continue: 0
Enter number of data elements to print:7
The doubly linked list is: 7    77      777     7777    7      77      777     Enter number of data elements to print in reverse:7
The reversed DLL is: 7777       777     77      7       7777    777     77
Process returned 0 (0x0)
```

DLL_insert_del_disp_all_in_one_char

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   char data;
   struct node *prev, *next;
}*st, *tp, *nn;
void delt()
{
char sel;
tp=st;
printf("Enter the value to delete:");
scanf("%s",&sel);
while (tp->data!=sel)
tp=tp->next;
nn=tp->next;
nn->prev=tp->prev;
tp->prev->next=nn;
free(tp);
}
void ins_at_end()
{
   nn=malloc(sizeof(struct node));
   printf("Enter data element to insert at end:");
```

```
    scanf("%s",&nn->data);
    tp->next=nn;
        nn->prev=tp;
        tp=tp->next;
        tp->next=NULL;
}
void ins_at_beg()
{
    tp=st;
    nn=malloc(sizeof(struct node));
    printf("Enter data element to insert at beg:");
    scanf("%s",&nn->data);
    nn->next=tp;
    tp->prev=nn;
    nn->prev=NULL;
    st=nn;
}
void create()
{
    int ch;
    do
    {
    nn=malloc(sizeof(struct node));
    printf("Enter the data element:");
    scanf("%s",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
        tp->prev=NULL;
    }
    else
    {
        tp->next=nn;

        nn->prev=tp;

        tp=tp->next;

        tp->next=NULL;

    }

    printf("Enter 0 to exit; any other value to continue:");

    scanf("%d",&ch);

    }while(ch!=0);

}

void ins_bef_el()

{

    nn=malloc(sizeof(struct node));

    int sel;

    printf("Enter element before which you wish to insert:");

    scanf("%s",&sel);

    printf("Enter data element to insert before %c:",sel);

    scanf("%s",&nn->data);

    tp=st;

    while (tp->data!=sel)

    tp=tp->next;

nn->prev=tp->prev;

nn->next=tp;

tp->prev->next=nn;

tp->prev=nn;

}

void display()

{

    int count=0;

    printf("The DLL is:");

    tp=st;

    while(tp!=NULL)

    {

        printf("%c\t",tp->data);
        tp=tp->next;
        count++;
    }
    printf("The number of elements in the DLL is:%d",count);
}
void revers()
{
    printf("\nThe revers DLL is:");
    tp=nn;
    while(tp!=NULL)
    {
        printf("%c->",tp->data);
        tp=tp->prev;
    }
}
int main()
{
    create();
    //ins_bef_el();
    //delt();
    display();
    revers();
    return 0;
}
```

DLL_insert_del_disp_all_in_one_float

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    float data;
    struct node *prev, *next;
}*st, *tp, *nn;
void delt()
```

```
{
float sel;
tp=st;
printf("Enter the value to delete:");
scanf("%f",&sel);
while (tp->data!=sel)
tp=tp->next;
nn=tp->next;
nn->prev=tp->prev;
tp->prev->next=nn;
free(tp);
}
void ins_at_end()
{
   nn=malloc(sizeof(struct node));
   printf("Enter data element to insert at end:");
   scanf("%f",&nn->data);
   tp->next=nn;
      nn->prev=tp;
      tp=tp->next;
      tp->next=NULL;
}
void ins_at_beg()
{
   tp=st;
   nn=malloc(sizeof(struct node));
   printf("Enter data element to insert at beg:");
   scanf("%f",&nn->data);
   nn->next=tp;
   tp->prev=nn;
   nn->prev=NULL;
   st=nn;
}
```

```
void create()
{
    int ch;
    do
    {
    nn=malloc(sizeof(struct node));
    printf("Enter the data element:");
    scanf("%f",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
        tp->prev=NULL;
    }
    else
    {
        tp->next=nn;
        nn->prev=tp;
        tp=tp->next;
        tp->next=NULL;
    }
    printf("Enter 0 to exit; any other value to continue:");
    scanf("%d",&ch);
    }while(ch!=0);
}
void ins_bef_el()
{
    nn=malloc(sizeof(struct node));
    float sel;
    printf("Enter element before which you wish to insert:");
    scanf("%f",&sel);
    printf("Enter data element to insert before %0.1f:",sel);
    scanf("%f",&nn->data);
    tp=st;
    while (tp->data!=sel)
    tp=tp->next;
nn->prev=tp->prev;
nn->next=tp;
tp->prev->next=nn;
tp->prev=nn;
}
void display()
{
    int count=0;
    printf("The DLL is:");
    tp=st;
    while(tp!=NULL)
    {
        printf("%0.1f\t",tp->data);
        tp=tp->next;
        count++;
    }
    printf("The number of elements in the DLL is:%d",count);
}
void revers()
{
    printf("\nThe revers DLL is:");
    tp=nn;
    while(tp!=NULL)
    {
        printf("%0.1f\t",tp->data);
        tp=tp->prev;
    }
}
int main()
{
    create();
    ins_bef_el();
    //delt();
    display();
    return 0;
}
```

DLL_insert_del_disp_all_in_one

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *prev, *next;
}*st, *tp, *nn;
void delt()
{
int sel;
tp=st;
printf("Enter the value to delete:");
scanf("%d",&sel);
while (tp->data!=sel)
tp=tp->next;
nn=tp->next;
nn->prev=tp->prev;
tp->prev->next=nn;
free(tp);
}
void ins_at_end()
{
    nn=malloc(sizeof(struct node));
    printf("Enter data element to insert at end:");
    scanf("%d",&nn->data);
    tp->next=nn;
```

```
        nn->prev=tp;
        tp=tp->next;
        tp->next=NULL;
}
void ins_at_beg()
{
    tp=st;
    nn=malloc(sizeof(struct node));
    printf("Enter data element to insert at beg:");
    scanf("%d",&nn->data);
    nn->next=tp;
    tp->prev=nn;
    nn->prev=NULL;
    st=nn;
}
void create()
{
    int ch;
    do
    {
    nn=malloc(sizeof(struct node));
    printf("Enter the data element:");
    scanf("%d",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
        tp->prev=NULL;
    }
    else
    {
        tp->next=nn;
        nn->prev=tp;
        tp=tp->next;
        tp->next=NULL;
    }
    printf("Enter 0 to exit; any other value to continue:");
    scanf("%d",&ch);
    }while(ch!=0);
}
void ins_bef_el()
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter element before which you wish to insert:");
    scanf("%d",&sel);
    printf("Enter data element to insert before %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while (tp->data!=sel)
    tp=tp->next;
nn->prev=tp->prev;
nn->next=tp;
tp->prev->next=nn;
tp->prev=nn;
}
void display()
{
    int count=0;
    printf("The DLL is:");
    tp=st;
    while(tp!=NULL)
    {
        printf("%d\t",tp->data);
        tp=tp->next;
        count++;
    }
    printf("The number of elements in the DLL is:%d",count);
}
void revers()
{
    printf("\nThe revers DLL is:");
    tp=nn;
    while(tp!=NULL)
    {
        printf("%d\t",tp->data);
        tp=tp->prev;
    }
}
int main()
{
    create();
    ins_bef_el();
    //delt();
    display();
    return 0;
}
```

insert_after_an_element_DLL_reduced

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do{
```

```
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
}
void insert_after_el()
{
   int sel;
   nn=malloc(sizeof(struct node));
   printf("Enter the selected element");
   scanf("%d",&sel);
   printf("Enter the data to insert after %d:",sel);
   scanf("%d",&nn->data);
   tp=st;
   while(tp->data!=sel)
      tp=tp->next;
   nn->prev=tp;
   nn->next=tp->next;
```

```
    tp->next->prev=nn;
    tp->next=nn;
    }
   void display()
{
tp=st;
printf("The doubly linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_after_el();
display();
return 0;
}
```

Circular_Doubly_linked_list_insert_after_el

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
```

```
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void insert_after_el()//same as that of DLL
{
  int sel;
  nn=malloc(sizeof(struct node));
  printf("Enter the selected element");
  scanf("%d",&sel);
  printf("Enter the data to insert after %d:",sel);
  scanf("%d",&nn->data);
  tp=st;
  while(tp->data!=sel)
    tp=tp->next;
    nn->prev=tp;
    nn->next=tp->next;
    tp->next->prev=nn;
    tp->next=nn;
    }
void display()
{
    int i,n;
    printf("Enter number of data elements to print:");
    scanf("%d",&n);
tp=st;
printf("The doubly linked list is: ");
for(i=0;i<n;i++)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
insert_after_el();
display();
return 0;
}
```

Circular_Doubly_linked_list_insert_before_el

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

```
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void ins_bef_el() //same as that of DLL
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter element before which you wish to insert:");
```

```
   scanf("%d",&sel);
   printf("Enter data element to insert before %d:",sel);
   scanf("%d",&nn->data);
   tp=st;
   while (tp->data!=sel)
   tp=tp->next;
nn->prev=tp->prev;
nn->next=tp;
tp->prev->next=nn;
tp->prev=nn;
}
void display()
{
   int i,n;
   printf("Enter number of data elements to print:");
   scanf("%d",&n);
tp=st;
printf("The doubly linked list is: ");
for(i=0;i<n;i++)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
create();
ins_bef_el();
display();
return 0;
}
```

Circular_Doubly_linked_list_insert_before_first_node

```
#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node *prev, *next;

}*st, *tp, *nn;

void create()

{

int ch;

do

{

nn=malloc(sizeof(struct node));

printf("Enter a data element: ");

scanf("%d",&nn->data);

if(st==NULL)

{

st=nn;

tp=nn;

tp->prev = NULL;

}

else

{

tp->next=nn;

nn->prev=tp;

tp=tp->next;

tp->next=NULL;

}

printf("Enter 0 to exit; any other value to continue: ");

scanf("%d",&ch);

}while(ch!=0);

nn->next=st;

st->prev=nn;
```

```
}
void ins_bef_el() //same as that of DLL; acts as insert at end when entered to insert a node before first node value;
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter element before which you wish to insert:");
    scanf("%d",&sel);
    printf("Enter data element to insert before %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while (tp->data!=sel)
    tp=tp->next;
nn->prev=tp->prev;
nn->next=tp;
tp->prev->next=nn;
tp->prev=nn;
}
void display()
{
    int i,n;
    printf("Enter number of data elements to print:");
    scanf("%d",&n);
tp=st;
printf("The doubly linked list is: ");
for(i=0;i<n;i++)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
int main()
{
```

```
create();
ins_bef_el();
display();
return 0;
}
```

Circular_Doubly_linked_list_display_once

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
```

```c
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void display()
{
tp=st;
printf("The Circular DLL is: ");
do{
printf("%d\t",tp->data);
tp=tp->next;
}while(tp->prev->data!=nn->data);
}
int main()
{
create();
display();
return 0;
}
```

Circular_Doubly_linked_list_print_data_at_given_position

```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
```

```
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void display()
{
printf("%d\t",st->next->next->next->next->data);//prints 5th element
}
int main()
{
create();
display();
return 0;
```

```
}
```

Circular_Doubly_linked_list_insert_at_beg

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
```

```
}while(ch!=0);

nn->next=st;

st->prev=nn;

printf("The element at 10th position is %d\t: ",st->next->next->next->next->next->next->next->next->next->data);//Extra line,just for practice

}

void ins_bef_el() //same as that of DLL (+2 new lines at end); acts as insert at beginning when entered to insert a node before first node value;

{

    nn=malloc(sizeof(struct node));

    int sel;

    printf("Enter element before which you wish to insert:");

    scanf("%d",&sel);

    printf("Enter data element to insert before %d:",sel);

    scanf("%d",&nn->data);

    tp=st;

    while (tp->data!=sel)

    tp=tp->next;

nn->prev=tp->prev;

nn->next=tp;

tp->prev->next=nn;

tp->prev=nn;

}

void display()

{

    int i,n;

    printf("Enter number of data elements to print:");

    scanf("%d",&n);

tp=st;

printf("The doubly linked list is: ");

for(i=0;i<n;i++)

{

printf("%d\t",tp->data);

tp=tp->next;
```

```
}
printf("The element at 10th position is %d\t: ",st->next->next->next->next->next->next->next->next->next->data);//Extra line,just for practice
}
int main()
{
create();
ins_bef_el();
display();
return 0;
}
```

print_at_position_CDLL_insert_at_beg

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
printf("The element at 10th position is %d\t: ",st->next->next->next->next->next->next->next->next->next->data);//Extra line,just for practice
}
void ins_bef_el() //same as that of DLL (+2 new lines at end); acts as insert at beginning when entered to insert a node before first node value;
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter element before which you wish to insert:");
    scanf("%d",&sel);
    printf("Enter data element to insert before %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while (tp->data!=sel)
    tp=tp->next;
nn->prev=tp->prev;
nn->next=tp;
tp->prev->next=nn;
tp->prev=nn;
}
void display()
{
    int i,n;
    printf("Enter number of data elements to print:");
    scanf("%d",&n);
tp=st;
printf("The doubly linked list is: ");
for(i=0;i<n;i++)
{
printf("%d\t",tp->data);
tp=tp->next;
}
printf("The element at 10th position is %d\t: ",st->next->next->next->next->next->next->next->next->next->data);//Extra line,just for practice
}
int main()
{
create();
ins_bef_el();
display();
return 0;
}
```

Print_alternate_elements in a SLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*st, *tp, *nn;
void creat()
{
    int ch;
    do{
        nn=malloc(sizeof(struct node));
    printf("Enter data element:");
    scanf("%d",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
    }
    else{
        tp->next=nn;
        tp=tp->next;
        tp->next=NULL;
    }
    printf("Enter 0 to exit; any value to continue:");
    scanf("%d",&ch);
    }while(ch!=0);
}
void disp_alt()
{
    tp=st;
    printf("The SLL is:");
    printf("%d",tp->data);
    tp=tp->next->next;
    while(tp!=NULL)
    {
    printf("->%d",tp->data);
    tp=tp->next->next;
    }
    }
int main()
{
    creat();
    disp_alt();
    return 0;
```

```
}
```

C Program to Search an Element in a Linked List

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*st, *tp, *nn;
void creat()
{
    int ch;
    do{
        nn=malloc(sizeof(struct node));
    printf("Enter data element:");
    scanf("%d",&nn->data);
    if(st==NULL)
    {
        st=nn;
        tp=nn;
    }
    else{
        tp->next=nn;
        tp=tp->next;
        tp->next=NULL;
    }
    printf("Enter 0 to exit; any value to continue:");
    scanf("%d",&ch);
    }while(ch!=0);
}
void disp_search()
{
    int sel,c=0;
```

```
   printf("Enter selection:");
scanf("%d",&sel);
   tp=st;
   while(tp!=NULL)
   {
      if(tp->data==sel)
      c++;
      tp=tp->next;
   }
   if(c==0)
   printf("The element is not available in SLL");
   else
   printf("The element is available in SLL");
   }
int main()
{
   creat();
   disp_search();
   return 0;
}
```

CDLL_dupli_check

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
```

```
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void dupli()
{
  int c;
  tp=st;
  while(nn!=st)
  {
    do
  {
    if(tp->data==nn->data)
      c++;
      tp=tp->next;
```

```
   }while(tp!=nn);
   nn=nn->prev;
   }
   if(c==0)
      printf("Duplicate elements not present in the CDLL");
   else
      printf("Duplicate elements present in the CDLL");
}
int main()
{
create();
dupli();
return 0;
}
```

CDLL_presence_of_data

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void presen()
{
  int sel,c;
  tp=st;
  printf("Enter search data: ");
  scanf("%d",&sel);
  do
  {
    if(tp->data==sel)
      c++;
      tp=tp->next;
  }while(tp->prev!=nn);
  if(c==0)
    printf("The element is not present in the CDLL");
  else
    printf("The element is present in the CDLL %d times",c);
}
```

```
int main()
{
create();
presen();
return 0;
}
```

CDLL_presence_of_data_using_exit

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *prev, *next;
}*st, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
tp->prev = NULL;
}
else
{
tp->next=nn;
nn->prev=tp;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
nn->next=st;
st->prev=nn;
}
void presen()
{
    int sel;
    tp=st;
    printf("Enter search data: ");
    scanf("%d",&sel);
    do
    {
        if(tp->data==sel)
        {
        printf("The element is present in the CDLL");
        exit(0);
        }
    tp=tp->next;
    }while(tp->prev!=nn);
   printf("The element is not present in the CDLL");
}
int main()
{
create();
presen();
return 0;
}
```

Print_allernative_elements_in_SLL

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*st, *tp,*nn;
void create()
{
    int ch,c=0;
    do{
        nn=malloc(sizeof(struct node));
        c++;
        printf("Enter data:");
        scanf("%d",&nn->data);
        if(st==NULL)
        {
            st=nn;
            tp=nn;
        }
        else
        {
            tp->next=nn;
            tp=tp->next;
            tp->next=NULL;
        }
        printf("Enter 0 to exit, any value to continue:");
        scanf("%d",&ch);
    }while(ch!=0);
    printf("The number of elements is %d",c);
    }
    void display()
    {
      tp=st;
      printf("The elements in the SLL are:");
       printf("%d",tp->data);
        tp=tp->next;
        if(tp==NULL)
        exit(0);
         tp=tp->next;
      while(tp!=NULL)
      {
         printf("->%d",tp->data);
         tp=tp->next;
      if(tp==NULL)
        exit(0);
         tp=tp->next;
      }
   }
   int main()
   {
      create();
    //  revers();
     display();
      return 0;
   }
```

Search_element_in_sll

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
}*st, *tp,*nn;
void create()
{
```

```
int ch,c=0;
do{
  nn=malloc(sizeof(struct node));
  c++;
  printf("Enter data:");
  scanf("%d",&nn->data);
  if(st==NULL)
  {
    st=nn;
    tp=nn;
  }
  else
  {
    tp->next=nn;
    tp=tp->next;
    tp->next=NULL;
  }
  printf("Enter 0 to exit, any value to continue:");
  scanf("%d",&ch);
}while(ch!=0);
  }
void display()
{
  int sel;
  printf("Enter search data:");
  scanf("%d",&sel);
  tp=st;
  while(tp!=NULL)
  {
  if(tp->data==sel)
  {
  printf("The element is present in SLL");
  exit(0);
```

```
        }
        tp=tp->next;
        }
        printf("The element is not present in SLL");
    }
    int main()
    {
        create();
        display();
        return 0;
    }
```

Concatenating_Singly_linked_lists

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*st, *stf, *tp, *nn;
struct node* create()
{
int ch;
st=NULL;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->next=nn;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");
scanf("%d",&ch);
}while(ch!=0);
return st;
}
void display(struct node* st)
{
tp=st;
printf("The linked list is: ");
while(tp!=NULL)
{
printf("%d\t",tp->data);
tp=tp->next;
}
}
struct node* concat(struct node* stf, struct node* st)
{
   tp=stf;
 while(tp->next!=NULL)
       tp=tp->next;
   tp->next=st;
 return stf;
}
int main()
{
stf=create();
st=create();
```

```
display(stf);
display(st);
stf=concat(stf, st);
display(stf);
return 0;
}
```

Singly_linked_list_split

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
int data;
struct node *next;
}*st, *stl, *tp, *nn;
void create()
{
int ch;
do
{
nn=malloc(sizeof(struct node));
printf("Enter a data element: ");
scanf("%d",&nn->data);
if(st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->next=nn;
tp=tp->next;
tp->next=NULL;
}
printf("Enter 0 to exit; any other value to continue: ");

scanf("%d",&ch);

}while(ch!=0);

}

void display(struct node* st)

{

tp=st;

printf("The linked list is: ");

while(tp!=NULL)

{

printf("%d\t",tp->data);

tp=tp->next;

}

}

void split()

{

   tp=st;

   int i,n;

   printf("Enter split position:");

   scanf("%d",&n);

   for(i=0;i<n-1;i++)

       tp=tp->next;

   stl=tp->next;;

   tp->next=NULL;

}

int main()

{

create();

display(st);

split();

display(st);

display(stl);

return 0;
```

```
}
```

CSLL_all_in_one_

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *next;
}*st,*tp,*nn;
void create_CSLL()
{
    int ch,c;
    do{
            c++;
        nn=malloc(sizeof(struct node));
        printf("Enter data:");
        scanf("%d",&nn->data);
        if(st==NULL)
        {
            st=nn;
            tp=nn;
          //  st->prev=NULL;
        }
        else
        {
            tp->next=nn;
     //       nn->prev=tp;
            tp=tp->next;
            tp->next=NULL;
        }
        printf("Enter 0 to exit; any value otherwise:");
        scanf("%d",&ch);
    }while(ch!=0);
```

```
 //   st->prev=nn;
    nn->next=st;
  //  printf("There are %d elements in CSLL",c);
}
void display()
{
    tp=st;
    printf("CSLL is:");
    do{
        printf("%d\t",tp->data);
        tp=tp->next;
    }while(tp!=st);
}
void ins_at_end()
{
    nn=malloc(sizeof(struct node));
    printf("Enter data to ins at end:");
    scanf("%d",&nn->data);
    tp->next=nn;
    tp=tp->next;
    tp->next=NULL;
    //st->prev=nn;
    nn->next=st;
    //st=nn;
    //nn=nn->prev;
    }
    void ins_after_el()
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter selected data:");
    scanf("%d",&sel);
    printf("Enter element to insert after %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while(tp->data!=sel)
        tp=tp->next;
    nn->next=tp->next;
    tp->next=nn;
    while(nn->next!=st)
        nn=nn->next;
}
void del_at_beg()
{
    tp=st;
    st=st->next;
    free(tp);
    nn->next=st;
}
void del_at_pos()
{
    int i,n;
 printf("Enter position:");
 scanf("%d",&n);
 tp=st;
 for(i=0;i<n-2;i++)
    tp=tp->next;
 nn=tp->next;
 tp->next=nn->next;
 free(nn);
 nn=tp;
 while(nn->next!=st)
    nn=nn->next;
}
int main()
{
    create_CSLL();
    //ins_at_end();
    //ins_after_el();
    //del_at_beg();
  //  del_at_pos();
    display();
    return 0;
}
```

CSLL Reverse

```
# include<stdio.h>
# include<stdlib.h>
struct node
{
    int dt;
    struct node *nt;
}*st, *nn, *tp;
int x;
void revers()
{
      nn=st;
    tp=st->nt;
    st=tp->nt;
    tp->nt=nn;
    nn->nt=NULL;
    while(tp->dt!=x)
    {
        nn=tp;
        tp=st;
        st=st->nt;
        tp->nt=nn;
    }
    st=nn;
    printf("The list is reversed: ");
```

```
}
void create()
{
     int ch;
    do
    {
nn = malloc(sizeof(struct node));
printf("Enter the data of new node: ");
scanf("%d", &nn->dt);
if (st==NULL)
{
st=nn;
tp=nn;
}
else
{
tp->nt=nn;
tp=tp->nt;
}
printf("Enter any non-zero value to add another node. Enter 0 otherwise: ");
scanf("%d",&ch);
}while(ch!=0);
nn->nt=st;
x=st->dt;
}
void disp()
{
    int i=0;
    printf("The data elements of the linked list are: ");
tp=st;
do{
    i++;
printf("%d\t",tp->dt);
tp=tp->nt;
}while (i<7);
}
int main ()
{
 create();
revers();
disp();
return 0;
}
```

CDLL_all_in_one

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
    int data;
    struct node *prev, *next;
}*st,*tp,*nn;
void create_CDLL()
{
    int ch,c;
    do{
            c++;
        nn=malloc(sizeof(struct node));
        printf("Enter data:");
        scanf("%d",&nn->data);
        if(st==NULL)
        {
            st=nn;
            tp=nn;
            st->prev=NULL;
        }
        else
        {
            tp->next=nn;
           nn->prev=tp;
            tp=tp->next;
            tp->next=NULL;
        }
        printf("Enter 0 to exit; any value otherwise:");
        scanf("%d",&ch);
    }while(ch!=0);
    st->prev=nn;
    nn->next=st;
        printf("There are %d elements in CDLL",c);
}
void display()
{
    tp=st;
    printf("CDLL is:");
    do{
        printf("%d\t",tp->data);
        tp=tp->next;
    }while(tp!=st);
}
void ins_at_end()
{
    nn=malloc(sizeof(struct node));
    printf("Enter data to ins at end:");
    scanf("%d",&nn->data);
    tp->next=nn;
    tp=tp->next;
    tp->next=NULL;
    st->prev=nn;
    nn->next=st;
```

```
   // st=nn;
  //  nn=nn->prev;
    }
    void ins_after_el()
{
    nn=malloc(sizeof(struct node));
    int sel;
    printf("Enter selected data:");
    scanf("%d",&sel);
    printf("Enter element to insert after %d:",sel);
    scanf("%d",&nn->data);
    tp=st;
    while(tp->data!=sel)
        tp=tp->next;
    nn->next=tp->next;
    nn->prev=tp;
    tp->next->prev=nn;
    tp->next=nn;
    while(nn->next!=st)
        nn=nn->next;
}
void del_at_beg()
{
    tp=st;
    st=st->next;
    free(tp);
    nn->next=st;
    st->prev=nn;
}
void del_at_pos()
{
    int i,n;
 printf("Enter position:");
 scanf("%d",&n);
 tp=st;
 for(i=0;i<n-2;i++)
    tp=tp->next;
 nn=tp->next;
 tp->next=nn->next;
 nn->next->prev=tp;
 free(nn);
 nn=tp;
 while(nn->next!=st)
    nn=nn->next;
}
void revers()
{
    tp=nn;
    printf("Reverse CDLL is:");
    do{
    printf("%d\t",tp->data);
    tp=tp->prev;
    }while(tp->next!=st);
    }
void el_search()
{
    int sel;
    tp=st;
    printf("Enter search data:");
    scanf("%d",&sel);
    while(tp->next!=st)
    {
        if(tp->data==sel)
    {
   printf("Element found");
   exit(0);
```

```
    }
        tp=tp->next;
}
printf("Element not found");
}
int main()
{
    create_CDLL();
   // ins_at_end();
    //ins_after_el();
    //del_at_beg();
   // del_at_pos();
  revers();
   // display();
    //el_search();
    return 0;
}
```