

the default keyword is used. purpose was solely not to disturb those classes which have already implemented the interface.

Double Linked list : (a) create list (b) display list (c) insert at the beginning (d) delete node

```
#include <stdio.h>
#include <stdlib.h>
struct node {
    struct node *next;
    int data;
    struct node *prev;
};
struct node *start=NULL;
struct node *create_ll (struct node * );
struct node *display (struct node * );
struct node *insert_beg (struct node * );
struct node *insert_before (struct node * );
struct node *delete_node (struct node * );
struct node *create_ll (struct node * start);
{
    struct node *newnode, *ptr;
    int num;
    printf ("In enter -1 to end");
    printf ("In Enter a value");
    while (num != -1)
    {
        if (start == NULL)
        {
            newnode = (struct node *) malloc (sizeof (struct node));
            newnode->data = num;
            newnode->next = NULL;
            newnode->prev = NULL;
            start = newnode;
        }
        else
        {
            newnode = (struct node *) malloc (sizeof (struct node));
            newnode->data = num;
            newnode->next = start;
            newnode->prev = NULL;
            start->prev = newnode;
            start = newnode;
        }
    }
}
```

```

newnode->prev = NULL;
newnode->data = num;
newnode->next = NULL;
start = newnode;
}

else
{
    ptr = start;
    newnode = (struct node*) malloc(sizeof(struct node));
    newnode->data = num;
    newnode->next = NULL;
    while (ptr->next != NULL)
    {
        ptr = ptr->next;
    }
    ptr->next = newnode;
    newnode->prev = ptr;
    newnode->next = NULL;
    start = newnode;
    return start;
}

printf ("In enter the data : ");
scanf ("%d", &num);
return start;
}

struct node *insert_before (struct node *start)
{
    struct node *ptr, *newnode;
    int num, val;
    printf ("In enter the data");
    scanf ("%d", &num);
    printf ("In enter the value before which data has to
            inserted");
    scanf ("%d", &val);
    newnode = (struct node*) malloc (sizeof(struct node));
    newnode->data = num;
    newnode->next = start;
    while (ptr->data != val)
    {
        ptr = ptr->next;
    }
    printf ("%.d", ptr->data);
}

```

```
prntf ("4. Display");
```

```
ptr = ptr -> next;  
}  
newnode -> next = ptr;  
newnode -> prev = ptr -> prev;  
ptr -> prev -> next = newnode;  
ptr -> prev = newnode;
```

```
return start;
```

```
}
```

```
struct node * delete_node (struct node * start)
```

```
{  
    struct node * ptr,
```

```
    int val,  
    print ("Enter value to be deleted");
```

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

```
    ptr = start,  
    while (ptr -> data != val)
```

```
        ptr = ptr -> next;
```

```
        ptr -> prev -> next = ptr -> next;
```

```
        ptr -> next -> prev = ptr -> prev;
```

```
        free (ptr);
```

```
        return start;
```

```
    }
```

~~char * display (struct node * start);~~

```
int main ()
```

```
{  
    int choice;
```

```
    prntf ("Menu");  
    prntf ("1. Create list");
```

```
    prntf ("2. Insert-before");
```

~~prntf ("3. Delete");~~