SOURCE CODE: DOUBLY LINKED LIST:

```
#include<stdio.h>

#include<conio.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_end(struct node*);

struct node *delete_beg(struct node*);

struct node *delete_end(struct node*);

int main()

{

        int option;

        //clrscr();

        do

        {

        printf("\n *****Main Menu*****\n1.Create a List\n2.Display the list\n3.Add a node in the beginning\n4.Add a node at the end\n5.Delete a node from the beginning\n 6:Delete a node from the end\n7.EXIT\nEnter your option\n");

        scanf("%d",&option);

        switch(option)

        {

                case 1:

                        start=create_ll(start);
```

```
                printf("LINKED LIST CREATED\n");
                break;
            case 2:
                start=display(start);
                break;
            case 3:
                start=insert_beg(start);
                break;
            case 4:
                start=insert_end(start);
                break;
            case 5:
                start=delete_beg(start);
                break;
            case 6:
                    start=delete_end(start);
                break;
        }
   }while(option!=7);
   getch();
   return 0;
}
struct node *create_ll(struct node *start)
{
        struct node *new_node;
   int num;
   printf("Enter the data\n");
   scanf("%d",&num);
   while(num!=-1)
```

```
    {
                if(start==NULL)
                {
                        start=(struct node*)malloc(sizeof(struct node*));
                        start->prev=NULL;
                        start->data=num;
                        start->next=NULL;
                }
                else
                {
                new_node =(struct node*)malloc(sizeof(struct node*));
                new_node->prev=NULL;
                new_node->data=num;
                new_node->next=start;
                start->prev=new node;
                start=new_node;
            }
            printf("Enter the data\n");
            scanf("%d",&num);
        }
        return start;
}
struct node *display(struct node *start)
{
    struct node *ptr;
    ptr=start;
    while(ptr!=NULL)
        {
                printf("\t %d",ptr->data);
```

```
                ptr=ptr->next;
        }
        return start;
}
struct node *insert_beg(struct node *start)
{
        struct node *new_node;
        int num;
        printf("Enter the data\n");
        scanf("%d",&num);
        new_node=(struct node *)malloc(sizeof(struct node *));
        start->prev=new_node;
        new_node->next=start;
        new_node->prev=NULL;
        new_node->data=num;
        start=new_node;
        return start;
}
struct node *insert_end(struct node *start)
{
        struct node *ptr,*new_node;
        int num;
        printf("Enter the data\n");
        scanf("%d",&num);
        new_node=(struct node *)malloc(sizeof(struct node *));
        new_node->data=num;
        ptr=start;
        while(ptr->next!=NULL)
                ptr=ptr->next;
```

```
        ptr->next=new_node;
        new_node->prev=ptr;
        new_node->next=NULL;
        return start;
}
struct node *delete_beg(struct node *start)
{
        struct node *ptr;
         ptr=start;
        start=start->next;
        free(ptr);
        return start;
}
struct node *delete_end(struct node *start)
{
        struct node *ptr,*preptr;
        ptr=start;
        while(ptr->next!=NULL)
        {
                preptr=ptr;
                ptr=ptr->next;
        }
        preptr->next=NULL;
        free(ptr);
        return start;
}
```

OUTPUT:

```
 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
1
Enter the data
10
Enter the data
20
Enter the data
30
Enter the data
-1
LINKED LIST CREATED
```

```
 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
2
         30      20      10
```

```
 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
3
Enter the data
5

 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
4
Enter the data
35

 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
5
```

```
 *****Main Menu*****
1.Create a List
2.Display the list
3.Add a node in the beginning
4.Add a node at the end
5.Delete a node from the beginning
6:Delete a node from the end
7.EXIT
Enter your option
6
```