```
/*Program for Doubly Linked List.*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void addstart();
void addmiddle();
void addlast();
void show();
void reverseshow();
void deldata();
void count();
void search();
void sort();

struct node
{
      int data;
      struct node *prev;
      struct node *next;
}*start=NULL,*last=NULL,*p,*ptr,*newnode;

int main()
{
      int ch;
      clrscr();
      do
      {
            printf("\n\n1. Add Start.\n");
            printf("2. Add Middle.\n");
            printf("3. Add Last.\n");
            printf("4. Show.\n");
            printf("5. Reverse Show.\n");
            printf("6. Delete.\n");
            printf("7. Count.\n");
            printf("8. Search.\n");
            printf("9. Sort.\n");
            printf("10. Exit.\n");
            printf("\nEnter your choice: ");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1: addstart();
                        break;
                  case 2: addmiddle();
                        break;
                  case 3: addlast();
                        break;
                  case 4: show();
                        break;
                  case 5: reverseshow();
                        break;
                  case 6: deldata();
```

```
                    break;
                case 7: count();
                    break;
                case 8: search();
                    break;
                case 9: sort();
                    break;
                case 10: exit(0);
                default: printf("\nYou have entered wrong choice!");
            }
      }while(ch!=10);
      return 0;
}

void addstart()
{
      newnode=(struct node *)malloc(sizeof(struct node));
      newnode->next=NULL;
      newnode->prev=NULL;
      printf("\n\nEnter the data that you want to Insert at start: ");
      scanf("%d",&newnode->data);
      if(start==NULL)
      {
            start=newnode;
            last=newnode;
      }
      else
      {
            newnode->next=start;
            start->prev=newnode;
            start=newnode;
      }
}

void addmiddle()
{
      int n;
      newnode=(struct node *)malloc(sizeof(struct node));
      newnode->next=NULL;
      newnode->prev=NULL;
      printf("\n\nEnter the data that you want to Insert at middle: ");
      scanf("%d",&newnode->data);
      if(start==NULL)
      {
            start=newnode;
            last=newnode;
      }
      else
      {
            printf("\n\nEnter the data after that you want to Insert:
");
            scanf("%d",&n);
```

```
            for(p=start;p!=NULL;p=p->next)
            {
                  if(p->data==n)
                  {
                        newnode->prev=p;
                        newnode->next=p->next;
                        p->next->prev=newnode;
                        p->next=newnode;
                        break;
                  }
             }
      }
}

void addlast()
{
      newnode=(struct node *)malloc(sizeof(struct node));
      newnode->next=NULL;
      newnode->prev=NULL;
      printf("\n\nEnter the data that you want to Insert at last: ");
      scanf("%d",&newnode->data);
      if(start==NULL)
      {
            start=newnode;
            last=newnode;
      }
      else
      {
            last->next=newnode;  /* or for(p=start;p!=NULL;p=p->next);
                                    p->next=newnode;*/
            newnode->prev=last;
            last=newnode;
      }
}

void show()
{
      printf("\n\nFollowing numbers are presented in the list:\n");
      for(p=start;p!=NULL;p=p->next)
      {
            printf("Value= %5d\t Address= %u\n",p->data,p);
      }
}

void reverseshow()
{
      printf("\n\nFollowing numbers are presented in the list:\n");
      for(p=last;p!=NULL;p=p->prev)
      {
            printf("Value= %5d\t Address= %u\n",p->data,p);
      }
}
```

```
void deldata()
{
      int n;
      if(start==NULL)
      {
            printf("\n\nDoubly Linked List is empty.");
      }
      else
      {
            printf("\n\nEnter the data that you want to delete:  ");
            scanf("%d",&n);
            for(p=ptr=start;p!=NULL;ptr=p,p=p->next)
            {
                  if(p->data==n)
                  {
                        if(p==start)
                        {
                              start=p->next;
                              start->prev=NULL;
                              free(p);
                              break;
                        }
                        else if(p==last)
                        {
                              last=ptr;
                              ptr->next=NULL;
                              free(p);
                              break;
                        }
                        else
                        {
                              ptr->next=p->next;
                              p->next->prev=ptr;
                              free(p);
                              break;
                        }
                  }
            }
      }
}

void count()
{
      int c=0;
      for(p=start;p!=NULL;p=p->next)
      {
            c++;
      }
      printf("\n\nTotal number of items present in the Doubly Linked
List= %d",c);
}
```

```
void search()
{
      int n,flag=0,c=0;
      printf("\n\nEnter the data that you want to search: ");
      scanf("%d",&n);
      for(p=start;p!=NULL;p=p->next)
      {
            c++;
            if(p->data==n)
            {
                  printf("\n\nData %d found at %dth position.",n,c);
                  flag=1;
                  break;
            }
      }
      if(flag==0)
      {
            printf("\n\nData not found in the list.");
      }
}

void sort()
{
      int temp;
      for(p=start;p!=NULL;p=p->next)
      {
            for(ptr=start;ptr!=NULL;ptr=ptr->next)
            {
                  if(ptr->data < ptr->next->data)
                  {
                        temp=ptr->data;
                        ptr->data=ptr->next->data;
                        ptr->next->data=temp;
                  }
            }
      }
}
```