//implementation of double linked list

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

struct node
{
   int data;
  struct node *prev,*next;
};

   struct node *head=NULL,*last=NULL;
   void create();
   void insert();
   void delet();
   void display();
   void search();

void create()
{
   struct node *temp;
   temp=(struct node*)malloc(sizeof(struct node));
   int n;
   printf("\nEnter an Element:");
   scanf("%d",&n);
   temp->data=n;
   temp->next=NULL;
   temp->prev=NULL;
   if(head==NULL)
   {
      head=temp;
      last=head;
   }
    else
```

```
    {
       last->next=temp;
       temp->prev=last;
       last=temp;
    }
}
void insert()
{
    struct node *old,*cur,*temp;
    old=NULL;
    cur=head;
    int count=1,pos,ch,n;
    temp=(struct node*)malloc(sizeof(struct node));
    printf("\nEnter an Element:");
    scanf("%d",&n);
    temp->data=n;
    temp->next=NULL;
    temp->prev=NULL;
    printf("\nINSERT AS\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN
FIRST&LAST NODES");
    printf("\nEnter Your Choice:");
    scanf("%d",&ch);
    switch(ch)
    {
    case 1:
       temp->next=head;
       head->prev=temp;
       head=temp;
       break;
    case 2:
       last->next=temp;
       temp->prev=last;
       last=temp;
       break;
    case 3:
```

```
        printf("\nEnter the Position to Insert:");
        scanf("%d",&pos);
        while(count!=pos)
        {
           old=cur;
           cur=cur->next;
           count++;
        }
        if(count==pos)
        {
          temp->next=old->next;
          cur->prev=temp;
          old->next=temp;
          temp->prev=old;
        }
        else
           printf("\nNot Able to Insert");
        break;

    }
}
void delet()
{
    struct node *old=NULL,*cur=head;
    int count=1,pos,ch;
    printf("\nDELETE\n1:FIRSTNODE\n2:LASTNODE\n3:IN BETWEEN
FIRST&LAST NODES");
    printf("\nEnter Your Choice:");
    scanf("%d",&ch);
    switch(ch)
    {
    case 1:
        if(head==NULL)
        {
                printf("\nNot Able to Delete");
```

```
      }
      else
      {
       printf("\nDeleted Element is %d",head->data);
      if(head==last)
      {
      head=last=NULL;
      }
      else
      {
      struct node *temp;
      temp=head;
      head=head->next;
      head->prev=NULL;
      }
    }
          break;
case 2:
      if(head==NULL)
      {
      printf("\nNot Able to Delete");
      }
      else
      {

    while(cur!=last)
    {
       old=cur;
       cur=cur->next;
    }
    if(cur==last)
    {
       printf("\nDeleted Element is: %d",cur->data);
       if(old==NULL)
       {
```

```
          head=NULL;
          }
          else
          {
          old->next=NULL;
          last=old;
          }
        }
       }
       break;
  case 3:
       printf("\nEnter the Position of Deletion:");
       scanf("%d",&pos);
         if(head==NULL)
         {
         printf("\nNot Able to Delete");
         }
         else
         {
         while(count!=pos)
         {
                        old=cur;
                cur=cur->next;
                count++;
                 }
         if(count==pos)
         {
                printf("\nDeleted Element is:%d",cur->data);
                old->next=cur->next;
                (cur->next)->prev=old;
         }
       }
       break;
    }
}
```

```
void display()
{
   struct node *temp=head;
   if(temp==NULL)
   {
      printf("\nList is Empty");
   }
   while(temp!=NULL)
   {
      printf("%d",temp->data);
      printf("-->");
      temp=temp->next;
   }
   printf("NULL");
}
void search()
{
   int value,pos=0;
   int flag=0;
   if(head==NULL)
   {
      printf("List is Empty");
      return;
   }
   printf("Enter the Value to be Searched:");
   scanf("%d",&value);
   struct node *temp;
   temp=head;
   while(temp!=NULL)
   {
      pos++;
      if(temp->data==value)
      {
         flag=1;
         printf("Element %d is Found at %d Position",value,pos);
```

```
            return;
        }
        temp=temp->next;
    }
    if(!flag)
    {
        printf("Element %d not Found in the List",value);
    }
}
int main()
{
    int ch;
    while(1)
    {
        printf("\n**** MENU ****");

printf("\n1:CREATE\n2:INSERT\n3:DELETE\n4:SEARCH\n5:DISPLAY\n6:EXI
T\n");
        printf("\nEnter Your Choice:");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:
            create();
            break;
        case 2:
            insert();
            break;
        case 3:
            delet();
            break;
        case 4:
            search();
            break;
        case 5:
```

```
            display();
            break;
        case 6:
            return 0;
        }
    }
    return 0;
}
```