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

struct node
{
  struct node *prev;
  char data;
  struct node *next;
};
struct node *start = 0;
struct node *CreateNode();
void InsertStart();
void InsertMiddle();
void InsertEnd();
void DeleteStart();
void DeleteMiddle();
void DeleteEnd();
void display();

int main()
{
  int ch;
  printf("\n 1. Insert at start.");
  printf("\n 2. Insert at middle.");
  printf("\n 3. Insert at end.");
  printf("\n 4. Delete The start.");
  printf("\n 5. Delete The middle one.");
  printf("\n 6. Delete The end.");
  printf("\n 7. Display.");
  printf("\n 8. Exit.");
  while(1)
  {
    printf("\n\n>>Enter your choice :");
    scanf("%d",&ch);
    (void)getchar();
    switch(ch)
    {
      case 1:
        InsertStart();
        break;
      case 2:
        InsertMiddle();
        break;
      case 3:
        InsertEnd();
        break;
      case 4:
        DeleteStart();
        break;
      case 5:
        DeleteMiddle();
        break;
      case 6:
        DeleteEnd();
        break;
      case 7:
        display();
        break;
      case 8:
        exit(0);
      default:
        printf("\n You have enterd a wrong input.");
    }
  }
  return 0;
}

struct node *CreateNode()
{
  struct node *cn;
  cn = (struct node*)malloc(sizeof(struct node));
  printf("\n Enter data :");
  scanf("%c",&cn->data);
  cn->prev = 0;
  cn->next=0;
  return cn;
}
void InsertStart()
{
  struct node *is;
  is = CreateNode();
  if(start == 0)
  {
    start = is;
  }
  else
  {
    is->next = start;
    start->prev = is;
    start = is;
  }
}
void InsertMiddle()
{
  struct node *im,*search;
  char data;
  im = CreateNode();
  if(start == 0)
  {
    start = im;
  }
  else
  {
    printf("\n-->Enter data which after you want to insert:");
    (void)getchar();
    scanf("%c",&data);
    search = start;
    while(search != 0 && search->data != data)
    {
      search = search->next;
    }
    if(search == 0)
    {
      printf("\n%c is not found.",data);
    }
    if(search -> next == 0)
    {
      printf("\nCannot apply this method , please try insert the method.");
      return;
    }
    im->next = search->next;
    search->next->prev = im;
    search->next = im;
    im->prev = search;
  }
}
void InsertEnd()
{
  struct node *ie,*search;
  ie = CreateNode();
  if(start == 0)
  {
    start = ie;
  }
  else
  {
    search = start;
    while(search->next != 0)
    {
      search = search->next;
    }
    search->next = ie;
    ie->prev = search;
  }
}

void DeleteStart()
{
  struct node *ds;
  if(start == 0)
  {
    printf("\n No node present for delete.");
    return;
  }
  else if(start->next == 0)
  {
    ds = start;
    start = 0;
    free(ds);
  }
  else
  {
    ds = start;
    start = start->next;
    start->prev = 0;
    ds->next = 0;
    free(ds);
  }
  printf("\n Deleted succesfully.");
}
void DeleteEnd()
{
  struct node *search,*de;
  if(start == 0)
  {
    printf("\nNo data present for delete.");
  }
  else if(start->next == 0)
  {
    DeleteStart();
  }
  else
  {
    search = start;
    while(search->next->next != 0)
    {
      search = search->next;
    }
    de = search->next;
    search->next = 0;
    de->prev = 0;
    free(de);
    printf("\nDeleted succesfully.");
  }
}
void DeleteMiddle()
{
  struct node *dm,*search;
  char data;
  if(start == 0)
  {
    printf("\n No data present to be delete.");
  }
  else if(start->next == 0)
  {
    DeleteStart();
  }
```

```c
  else
  {
    printf("\n--> Enter the data that you want to delete :");
    scanf("%c",&data);
    if(start->data == data)
    {
      printf("\nCannot apply this method , please apply delete the start method.");
      return;
    }
    search = start;
    while( search->next != 0 && search->next->data != data)
    {
      search = search->next;
    }
    if(search ->next == 0)
    {
      printf("\n%c is not found.",data);
      return;
    }
    dm = search->next;
    if(dm->next == 0)
    {
      printf("\nCannot apply this method,please apply delete the end method.");
      return;
    }
    search->next = dm->next;
    dm->next->prev = search;
    dm->prev = 0;
    dm->next=0;
    free(dm);
    printf("\nDeleted succesfully");
  }
}

void display()
{
  struct node *d;
  if(start ==0)
  {
    printf("\n There are no any data to be print.");
  }
  else
  {
    d = start;
    printf("\nFrom start to end : [");
    while(d->next != 0)
    {
      printf("%c ",d->data);
      d = d->next;
    }
    printf("%c ]",d->data);
    printf("\nFrom end to start : [");
    while(d->prev != 0)
    {
      printf("%c ",d->data);
      d=d->prev;
    }
    printf("%c ]",d->data);
  }
}
```