```c
#include<stdio.h>
#include<stdlib.h>
struct node
{
  int data;
  struct node *next;
};
struct node *start;
struct node *createnode();
void InsertStart();
void InsertMiddle();
void InsertEnd();
void DeleteStart();
void DeleteMiddle();
void DeleteEnd();
void display();

int main()
{
  int ch;
  printf("\n1. Insert at start.");
  printf("\n2. Insert at middle.");
  printf("\n3. Insert at end.");
  printf("\n4. Delete from start.");
  printf("\n5. Delete from middle.");
  printf("\n6. Delete from end.");
  printf("\n7. Display.");
  printf("\n8. Exit.");
  choice:
  printf("\n\n>>Enter your choice:");
  scanf("%d",&ch);
  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("\nWrong input ! Try again.");
  }
  goto choice;
  return 0;
}

struct node *createnode()
{
  struct node *cn;
  cn = (struct node*)malloc(sizeof(struct node));
  printf("\nEnter data:");
  scanf("%d",&cn->data);
  cn->next=0;
  return cn;
}

void InsertStart()
{
  struct node *is,*search;
  is=createnode();
  if(start == 0)
  {
    start = is;
    is->next = is;
  }
  else
  {
    is->next = start;
    search = start;
    while(search->next != start)
    {
      search = search->next;
    }
    search->next = is;
    start = is;
  }
}
void InsertMiddle()
{
  struct node *im,*search;
  int data;
  im = createnode();
  if(start == 0)
  {
    start = im;
    im->next = start;
  }
  else
  {
    printf("\nEnter the dada which after that you want add :");
    scanf("%d",&data);
    search=start;
    while(search->data != data)
    {
      search = search->next;
      if(search == start)
      {
        printf("\n%d is not found !",data);
        return;
      }
    }
    im->next = search->next;
    search->next = im;
  }
}
void InsertEnd()
{
  struct node *ie,*search;
  ie = createnode();
  if(start == 0)
  {
    start = ie;
    ie->next = start;
  }
  else
  {
    search = start;
    while(search->next != start)
    {
      search = search->next;
    }
    search->next = ie;
    ie->next = start;
  }
}
void DeleteStart()
{
  struct node *ds,*search;
  if(start == 0)
  {
      printf("\nNo node aviable to delete.");
  }
  else if(start == start->next)
  {
      ds=start;
      start=0;
      free(ds);
  }
  else
  {
      search=start;
      while(search->next != start)
      {
          search=search->next;
      }
      ds=start;
      start=start->next;
      search->next = start;
      ds->next = 0;
      free(ds);
  }
  printf("\nDeleted Succesfully.");

}
void DeleteMiddle()
{
  struct node *dm,*search;
  int data;
  if(start==0)
  {
      printf("\nNo data aviable to delete.");
  }
  else
  {
      printf("\nEnter data thats you want to delete:");
      scanf("%d",&data);
      if(data == start->data)
      {
          printf("\nSorry! Cannot delete . Please try delete from start method.");
          return;
      }
      search=start;
      while(search->next->data != data)
      {
          search=search->next;
          if(search==start)
          {
              printf("\n%d is not found.",data);
              return;
          }
      }
      dm=search->next;
      search->next = dm->next;
      dm->next = 0;
      free(dm);
      printf("\nDeleted Succesfully.");
  }
}
void DeleteEnd()
{
  struct node *de,*search;
  if(start == 0)
  {
      printf("\nNo data aviable to delete.");
  }
  else if(start->next == start)
  {
      de = start;
      start = 0;
      free(de);
  }
  else
  {
      search=start;
      while(search->next->next != start)
      {
          search=search->next;
      }
      de=search->next;
      search->next = de->next;
      de->next = 0;
      free(de);
  }
  printf("\nDeleted Succesfully.");
}
void display()
{
  struct node *d;
  if(start == 0)
  {
   printf("\nNo data aviable to print");
  }
  else
  {
   d = start;
   printf("\nYour data = [");
   do
   {
      printf("%d ",d->data);
      d=d->next;
   }
   while(d != start);
   printf("]");
  }
}
```