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

void insert_beg();
void insert_at_end();
void insert_at_pos();
void delete_at_beg();
void delete_at_end();
void delete_at_pos();
void display();

struct node
{
 int data;
 struct node *next;
};
struct node *newnode,*temp,*temp1,*last,*mid;
struct node *head=NULL;
main()
{
 int choice;
 while(1)
 {
 printf("\nEnter choice :\n1 Insert at beginning \n2 Insert at end");
 printf("\n3 Insert at a position \n4 Delete from beginning \n5 Delete at end");
 printf("\n6 Delete from position   \n7 Display \n8 Exit\n");
 scanf("%d",&choice);
 switch(choice)
  {
   case 1:insert_beg();break;
   case 2:insert_at_end();break;
   case 3:insert_at_pos();break;

   case 4:delete_at_beg();break;
   case 5:delete_at_end();break;
   case 6:delete_at_pos();break;
   case 7:display();break;
   case 8:exit(0);
  }
 }
}

void insert_beg()
{
 newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter data into new node:");
 scanf("%d",&newnode->data);
 newnode->next=NULL;
 if(head==NULL)
 {
  head=newnode;
  newnode->next=newnode;
 }
 else
 {
  newnode->next=head;
  temp=head;
  while(temp->next!=head)
  {
   temp=temp->next;
  }
  temp->next=newnode;
  head=newnode;
 }
}

void insert_at_end()
{
 newnode=(struct node*)malloc(sizeof(struct node));

 printf("\nEnter data into new node:");
 scanf("%d",&newnode->data);
 newnode->next=NULL;
 if(head==NULL)
 {
  head=newnode;
  newnode->next=newnode;
 }

 else
 {
  temp=head;
  while(temp->next!=head)
  {
   temp=temp->next;
  }
  temp->next=newnode;
  newnode->next=head;
 }
}

void insert_at_pos()
{
 int i,pos;
 newnode=(struct node*)malloc(sizeof(struct node));
 printf("\nEnter data into new node:");
 scanf("%d",&newnode->data);
 newnode->next=NULL;
 printf("\nEnter position :");
 scanf("%d",&pos);
 if(pos==0)
 {
  newnode->next=head;
  temp=head;
  while(temp->next!=head)

  {
   temp=temp->next;
  }
  temp->next=newnode;
  head=newnode;
 }
 else
 {
  temp=head;
  for(i=0;i<pos-1;i++)
  {
   temp=temp->next;
   if(temp==head)
   {
     printf("\nPosition not found");
     return;
   }
  }
  newnode->next=temp->next;
  temp->next=newnode;
 }
}

void delete_at_beg()
{
 if(head==NULL)
 {
  printf("\n Linked list is empty");
 }
 else if(head->next==head)
 {
  head=NULL;
  printf("\nNode deleted successfully");
 }

 else
 {
  temp1=head;
  temp=head;
  while(temp->next!=head)
   temp=temp->next;
  head=head->next;
  temp->next=head;

  free(temp1);
  printf("\nNode deleted successfully");
 }
}

void delete_at_end()
{
 if(head==NULL)
 {
  printf("\n Linked list is empty");
 }
 else if(head->next==head)
 {
  head=NULL;
  printf("\nNode deleted successfully");

 }
 else
 {
  temp=head;
  while(temp->next->next!=head)
  {
   temp=temp->next;
  }
  last=temp->next;
  temp->next=head;
  printf("\nNode deleted successfully");

  free(last);
 }
}

void delete_at_pos()
{
 int i,pos;
 printf("\nEnter position to delete node:");
 scanf("%d",&pos);
 if(head==NULL)
 {
  printf("\n Linked list is empty");
 }
 else if(pos==0)
 {
  temp=head;
  temp1=head;
  while(temp->next!=head)
   temp=temp->next;

  temp->next=head->next;
  head=head->next;
  free(temp1);
  printf("\nNode deleted successfully");
 }
 else
 {
  temp=head;
  for(i=0;i<pos-1;i++)
  {
   temp=temp->next;
   if(temp==head)
   {
   printf("\nInvalid position");
   return;

   }

  }

  mid=temp->next;
  temp->next=temp->next->next;
  free(mid);
  printf("\nNode deleted successfully");
 }
}

void display()
{
 if(head==NULL)
 {
  printf("\nLinked list is empty");
 }
 else
 {
  temp=head;
  printf("\n Nodes in the linked list are :\n");
  while(temp->next!=head)
  {
   printf("%d => ",temp->data);
   temp=temp->next;
  }
  printf("%d ",temp->data);
 }
}
```