```
#include<stdio.h>
#include<stdlib.h>
int pos,ch,cho;
struct node{
  int data;
  struct node* next;
}*start=NULL;
struct node* current=NULL;
void add_node()
{

   struct node * nn;
   nn=(struct node*)malloc(sizeof(struct node*));
   nn->next=NULL;
   printf("Enter the data:");
   scanf("%d",&nn->data);
   if(start==NULL)
   {
      start=nn;
      current=nn;
   }
   else
   {
      current->next=nn;
      current=nn;
   }

}
void delet_beg()
{
   struct node * temp=start;
```

```
    if(start==NULL)
    {
      printf("Empty!!!!");
      return;
    }
    else
    {
       start=start->next;
       free(temp);
    }

}
void delet_end()
{
    struct node * temp=start;

    if(start==NULL)
    {
     printf("Empty!!!!");
      return;
    }
    else
    {
       while(temp->next!=current){
          temp=temp->next;
       }
       temp->next=NULL;
       free(current);
       current=temp;
    }
```

```
}
void delet_pos(pos)
{
   struct node * temp=start;


   if(start==NULL)
   {
      printf("Empty!!!!");
     return;
   }
   else
   {   struct node*temp=start;
      struct node* temp1;
      struct node* temp2;
      if(pos==0)
     {
      delet_beg();
     }
      for(int i=1;i<pos-2;i++)
         temp=temp->next;
      temp1=temp->next;
      temp2=temp1->next;
      temp->next=temp2;
      free(temp1);
   }
}
void display()
{
   if(start==NULL)
   {
      printf("Linked List is empty!!");
```

```
    return;
  }
  struct node*temp=start;
  while(temp!=NULL)
  {
    printf("%d ->",temp->data);
    temp=temp->next;

  }
  printf("NULL");
}
void main()
{
  while(1)
  {

  printf("\n 1.Add node\n 2.Delete at Beginning\n 3.Delete at End\n 4.Delete at Position\n
5.Display\n 6.Exit\n Enter your CHOICE:");
  scanf("%d",&ch);
  switch(ch)
  {
    case 1:add_node();
          break;
    case 2:delet_beg();
          break;
    case 3:delet_end();
          break;
    case 4:printf("Enter the position to delete:");
         scanf("%d",&pos);
         delet_pos();
```

```
            break;
        case 5:display();
               break;
        case 6:printf("Exiting Program.....");
               break;
        default:printf("INVAILD CHOICE!!");
    }

}
}
```

OUTPUT:

```
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:1
Enter the data:34

 1.Add node
 2.Delete at Beginning
 3.Delete at End
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:55
INVAILD CHOICE!!
 1.Add node
 2.Delete at Beginning
 3.Delete at End
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:1
Enter the data:32

 1.Add node
 2.Delete at Beginning
 3.Delete at End
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:4
Enter the position to delete:1

 1.Add node
 2.Delete at Beginning
 3.Delete at End
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:5
34 ->NULL
 1.Add node
 2.Delete at Beginning
 3.Delete at End
 4.Delete at Position
 5.Display
 6.Exit
 Enter your CHOICE:6
Exiting Program.......
```