// doubly linked list

```
Struct node{ Int data
             Struct node *next;
             Struct node * prev;
           };
   Struct node *head = NULL;
Void insert_beg()
{
     Struct node * new_node;
     newnode = (Struct node *) malloc (size of (Struct node));
     Scanf("%d", &newnode ->data);
      newnode -> next = NULL;
      newnode -> prev = NULL;
      if (head == NULL)
                head = newnode;
      else {
              newnode -> next = head.
              head -> prev = newnode;
              head = newnode
  }      }

 Void insert_end()
 {
     Struct node* newnode, *temp;
     newnode = (Struct node*) Malloc (size of (Struct node));
      scanf("%d", &newnode ->data);
      newnode -> next = NULL;
      Newnode -> prev = NULL;
      if(head == NULL)
             head = newnode;
      else { temp = head;
             While(temp -> next != NULL)
                 temp = temp -> next;
             temp -> next = newnode;
              newnode -> prev = temp;
           }
void insert_between()
     {
         int Listele;
         Struct node *newnode, *temp;
         Scanf("%d", &Listlle);
         newnode = (Struct node*) malloc (sizeof (Struct node));
          scanf("%d", &newnode-> data);
           newnode -> next = NULL;
           newnode -> prev = NULL;
         if(head == NULL)
             { printf("Empty List\n"); return;
             }
      temp=head
      while (temp->data!=listele)
         {
             temp=temp->next;
              if (temp = NULL)
                { printf("element not in list");
                   return;
                }
         }
    if (temp->next==NULL)
     {
        newnode->next= temp->next;
         temp->next = newnode;
          newnode->prev = temp;
           return;
     }
   newnode->next = temp->next;
   temp->next= newnode;
    newnode->prev=temp;
     return;
    }
   newnode->next= temp->next;
    temp->next=newnode;
   newnode->prev = temp;
    newnode->next->prev= newnode;
}
void del()
  {
          struct node *temp;
           int ele;
          if(head==NULL){ printf("Empty list\n"); return; }
           printf(" enter element to be deleted\n");
            scanf("%d",&ele);
              temp=head;
             While(temp->data!=ele)
             {
                 temp= temp->next;
                   if (temp==NULL){ printf("element not found\n");
                                      break; }
               }
            if(temp == head)
                      head=head->next;
            else if (temp->next==NULL){
                          temp= temp->prev;
                            temp->next=NULL; }
               Else {
                 temp->prev->next= temp->next;
                 temp->next->prev = temp->prev;
                }
   }
```

```
void display()
{
    struct node* temp;
     temp=head;
     while(temp!=NULL)
        {  printf("%d\t", temp->data);
             temp= temp->next;
        } printf(" \n");
     }

int main()
{
     int choice
      while(1){
              printf(" 1. Insert at beg \t  2. Insert at end \t
               3. insert after a no. \t 4. delete 5. Display 6. exit \n
                  enter your choice \n");
              scanf("%d", &choice
               switch(choice)
              {  case 1: insert_beg(); break;
                 case 2: insert_end(); break;
                 case 3: ~~insert after~~ insert_between(); break
                 case 4: del(); break;
                 case 5: display(); break
                 case 6: exit(0);
      }      }
    return 0;
}
```