DEV KANKAL
1BM19CS046

```
# include <stdio.h>
# include <stdlib.h>
void create( );
void display();
void dyine(int);
void insert - before ();
   struct node
{ int  data;
 Struct node * next;
 };
 Struct node * head = NULL
int main (int argc , char ** argv)
  {
   int choice, ele ;
     char ch;
  do
  { printf ("\n1. Create \n2. Display \n3. Delete
      \n4. Insert_before \n");
  Printf ("\n enter your choice : ");
   Scanf ("%d ,& choice );
   switch (choice)
  { case 1 : Create (); break ;
   Case2 : display (); break ;
  case 3 : printf (" enter the element to be deleted \n");
```

Scanned with CamScanner

```
scanf("%d",&ele);
 del fun(ele); break;
 case 4 : insert_before;
          break;
 }
   printf("\nDo you want to continue(y||Y):");
 fflush(stdin);
  scanf("%c",&ch);
 }while(ch=='y'|| ch=='Y');
    }
    void create()
    {
    struct node *newnode,*temp;
    int item;
  newnode=(struct node*)malloc(size of(struct node));
  printf("enter the data:");
  scanf("%d",&item);
  newnode->data=item;
   if(head==NULL)
   {
    newnode->next=NULL;
   head=newnode;
  printf("~~node~~ Node created\n");
   }
   else
```

Scanned with CamScanner

```
{
 temp= head ;
   while (temp->next! = NULL)
     {
       temp= temp -> next;
     }
     temp -> next= new node
     new node-> next = NULL;
     Printf("Node Created \n");

     } }

     void display()
      {
       Struct node* ptr = NULL;
         Ptr= head;
        if (ptr = NULL)
        { printf("Nothing to print \n");

        } else
         { while (ptr!= NULL)
          { printf ("%d", ptr ->data);
            Ptr = ptr-> next;
            }
            }
            }
```

Scanned with CamScanner

```
void del fun(int ele)
 {
  Struct node *temp ,*del = NULL;
    if(head==NULL)
     {
       printf("Empty list can't delete\n");
    return; }
  temp= head;
  if(head -> data == ele)
   {
    head= head -> next;
     return;
   } while( temp ->next != NULL)
   { if (temp -> next -> data == ele)
     { del = temp -> next;
      if (del -> next == NULL)
       temp -> next = NULL;
     else
     temp -> next = del -> next;
   } else
       temp= temp -> next;
   } if(del == NULL)
   { printf("element not found in the list\n");
      return;
    }}
```

Scanned with CamScanner

```
Void insert_begin ( )
 { struct node *new node ;
   int ele ;
   printf ("enter the element:");
   scanf ("%d",&ele);
   newnode= (struct node*)malloc(size of(struct
            node));
   newnode -> data =ele ;
   newnode -> next = head;
   head= new node ;
   }
```

Scanned with CamScanner