IBM19CS054
Dinyershan. K

Lab 6 :-

```
#include <stdio.h>
#include <stdlib.h>
void create();
void display();
void delete_front();
void delete_last();
void delete_at_node(int)
void insert_at before();
struct node.
{
  int data;
  struct node* next;
};
struct node* head=NULL;
int main()
{
  printf("\n1. Insert \n2. Display
        \n3. Delete at first \n4. Delete
        from end \n5. Delete at particular
        node \n6. Exit \n");
  printf("\n Enter your choice:");
  scanf("%d", &choice);
  switch(choice)
  {
  case1 : create(); break;
  case2 : display(); break;
  case3 : delete_front(); break.
  case4 : delete_front(); break.
  case5 : printf("Enter the element to
              be deleted \n");
    scanf("%d", &ele);
    delete_at_node(ele); break;
    default : exit(0);
  }
} while (choice==1 || choice==2 || choice==3
      || choice==4 || choice==5);
```

```
return 0;
}
void create()
{
struct node *newnode, *temp;
int item;
newnode = (struct node *) temp; int item;
    newnode = (struct node *) malloc
                (sizeof (struct node));
  printf("\n Enter the data: ");
  scanf("%d", &item);
  newnode->data = item;
  if (head == NULL)
{  newnode->next = NULL;
   head = newnode;
   printf("node created\n");
}
  else { temp = head; //traversing
  while (temp->next != NULL
  {
    temp = temp->next;
  } temp->next = newnode;
    newnode->next = NULL;
    printf("Node created \n");
  }
}
void delete_front()
{
  if (head == NULL)
{
  printf("Empty list. Can't delete \n");
  return;
} else
  head = head->next;
}
void delete_last()
{
struct node *temp;
```

```
if (head == NULL)
{ printf (" Empty list. Cant delete node");
  return; } else
  {
    temp = head;
    while ( temp -> next -> next != NULL)
    {
      //printf ("%d", temp -> data);
       temp = temp -> next
    } temp -> next = NULL;
    }
    }
void delete_at_node(int ele)
{
  struct node * temp, * del = NULL;
  if (head == NULL)
{ printf(" Empty list. Cant 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; // deleting at end
      else
      temp -> next = del -> next;
    }
      else
```

```
    temp = temp -> next;
  }
  if (del == NULL)
  {
    printf("Elements not found in the list \n");
    return;
  }
}
void display()
{
  struct node *ptr = NULL;
  ptr = head;
  if (ptr == NULL)
  {
    printf("list empty !!! \n");
  }
  else
  {
    while (ptr != NULL)
    {
      printf("list empty !!! \n");
    else { while (ptr != NULL)
      { printf("%d", ptr -> data);
        ptr = ptr -> next;
      }
    }
    printf("\n");
  }
```