DEV KANNAN
1BM19CS046
20/11/2020

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

struct node
{
  int data;
  struct node *next;
};
struct node * head=NULL;
void display();
void delfun();
void insert();
int main()
{ int ch, ele;
 do
 {
printf("\n 1.Create and insert \n 2. Display \n 3. Delete
      \n 4. Exit");
printf("\n Enter your choice:");
scanf("%d",&choice);
  switch(ch)
  { case 1 : insert() : break;
    case 2 : display() ; break;
    case 3 : delfun() : break;
     default : if(ch!=4)
      printf("invalid option");
  } while(ch!=5); }
```

```
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; }
}
void delfun()
{ struct node *temp; *del = NULL;
if (head == NULL)
{ printf("Empty list. Can't delete\n");
return; }
temp = head;
int ch, ele;
printf("Delete at\n1. Front\n2. Back\n3. Desired
element\n Enter choice:");
scanf("%d", &ch);
switch(ch)
{ case 1: del = head;
head = head -> next;
printf("Node deleted\n");
break; (2)
```

```
case 2: while (temp -> next -> next != NULL)
        { temp = temp -> next ; }
        del = temp -> next ;
        temp -> next = NULL;
        printf ("Node deleted \n");
        break;
case 3 : printf ("Enter the element to delete \n");
        scanf ("%d", &ele);
        if (head -> data == ele)
        { head = head -> next ; }
        else
        while (temp -> next != NULL)
        { if (temp -> next -> data == ele)
          { del = temp -> next ;
            if (del -> next == NULL)
              temp -> next = NULL;
            else
              temp -> next = del -> next ;
          }}
        if (del == NULL)
        { printf (" element not found in the list \n");
        return; }}
Void insert ()
{ struct node * temp, * newnode;
```

```
int item;
new node = (struct node *)malloc(sizeof(struct node));
scanf("%d", &item);
new node -> data item;
if (head == NULL)
{ head = new node; }
else
{ temp = head;
while (temp -> next != NULL)
{ temp = temp -> next; }
temp -> next = new node;
printf("\n Node created \n");
}}
```