DAYANAND

1BM19CS043

```c
#include <stdio.h>
#include <stdlib.h>
void create ();
void display ();
void define (int);
void insert_before ();
struct node
{
    int data;
    struct node *next;
};
struct node *head = NULL
int main (int arg c, @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 (" 1-d", &choice);
        switch (choice)
        {
            case 1: create (); break;
            case 2: display () ; break;
            case 3; printf ("enter the element to be deleted\n
```

```c
scanf("%d", &ele);

del_fun(ele); break;

case 4:  insert - before;
         break;
}
    printf("\n Do 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(sizeof(struct node));
    printf(" Enter the data :");
    scanf("%d", &item);
    newnode -> data = item;
    if(head == NULL)
    {
        newnode -> next = NULL;
        head = newnode;
        printf(" Node created\n");
    }
    else
```

```c
{   temp = head;
    while (temp -> next != NULL)
    {
        temp = temp -> next;
    }
    temp -> next = newnode
    newnode -> 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;
        }
    }
}
```

```c
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
     } }
```

```c
Void    insert _ before ()
{   struct    node  * newnode;
    int elu;
    Printf(" Enter   the element : ");
     scanf(" %d", &elu);
      newnode = (struct    node *) malloc (size of (struct
       node)) ;

      newnode -> data = elu;
      newnode -> next = head;
       head = newnode;
       y
```