Assignment 3:

- Write functions to implement the following operations on Circular Linked List.

  i. Create a circular linked list with a finite number of elements.
  ii. Insert an element at the (beginning & end) of the list.
  iii. Delete an element from the (beginning & end) of the list.
  iv. Traverse and print the content of the list.

Program:

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

struct node {
    int num;
    struct node * nextptr;
}*stnode;

struct node *tail,*p,*q,*store;

void ClListcreation(int n);
void ClLinsertNodeAtBeginning(int num);
void ClLinsertNodeAtEnd(int num);
void ClListDeleteFirstNode();
void ClListDeleteLastNode();
void displayClList();

int main()
{
    int n,num1,a,insPlc,item;
    stnode = NULL;
     //switch to display menu
     while(1)
     {
          printf("1.Create\n2.Traverse\n3.Insert 
First\n4.Insert Last\n5.Delete First\n6.Delete 
Last\n0.Exit\nYour Choice: ");
          scanf("%d",&a);
          switch(a)

          {
               case 1:
                    printf("\nEnter the number of nodes: 
");
                    scanf("%d",&n);
                    ClListcreation(n);
                    break;

               case 2:
                       displayClList();
                       break;
               case 3:
                       printf("\nEnter the information 
for the node to be inserted: ");
                       scanf("%d",&item);
                       ClinsertNodeAtBeginning(item);
                       break;

               case 4:
                       printf("\nEnter the information 
for the node to be inserted: ");
                       scanf("%d",&item);
                       ClinsertNodeAtEnd(item);
                       break;

               case 5:
                       ClListDeleteFirstNode();
                       break;

               case 6:
                       ClListDeleteLastNode();
                       break;
               case 0: exit(0);
               default:
                       printf("\nWrong input. Please try 
again...");
          }
     }
    return 0;
}

void ClListcreation(int n)
{
    int i, num;
    struct node *preptr, *newnode;

    if(n >= 1)
    {
        stnode = (struct node *)malloc(sizeof(struct
node));

        printf(" Input data for node 1 : ");
        scanf("%d", &num);
        stnode->num = num;
        stnode->nextptr = NULL;
        preptr = stnode;
        for(i=2; i<=n; i++)
        {
            newnode = (struct node *)malloc(sizeof(struct
node));
            printf(" Input data for node %d : ", i);
            scanf("%d", &num);
            newnode->num = num;
            newnode->nextptr = NULL;    // next address
of new node set as NULL
            preptr->nextptr = newnode;  // previous node
is linking with new node
            preptr = newnode;           // previous node
is advanced
        }
        preptr->nextptr = stnode;       //last node is
linking with first node
    }
}

void ClLinsertNodeAtBeginning(int num)
{
    struct node *newnode, *curNode;
    if(stnode == NULL)
    {
        printf(" No data found in the List yet.");
    }
    else
    {
        newnode = (struct node *)malloc(sizeof(struct
node));
        newnode->num = num;
        newnode->nextptr = stnode;
        curNode = stnode;
        while(curNode->nextptr != stnode)
        {
            curNode = curNode->nextptr;
        }
        curNode->nextptr = newnode;
        stnode = newnode;
    }
}

void ClLinsertNodeAtEnd(int num1)
{
          int a;
          a=num1;
          struct node *temp=(struct
node*)malloc(sizeof(struct node));
          temp->num=a;
          p=stnode;
          while(p->nextptr!=stnode)
          {
               p=p->nextptr;
          }
          p->nextptr=temp;
          temp->nextptr=stnode;
}

void ClListDeleteFirstNode()
{
          p=stnode;
          while(p->nextptr!=stnode)
          {
               p=p->nextptr;
          }
          store=stnode;
          stnode=stnode->nextptr;
          printf("\n The deleted node is -> %d\n",store-
>num);
```

```
          p->nextptr=stnode;
          free (store);
}

void ClListDeleteLastNode()
{
          p=stnode;
          while(p->nextptr!=stnode)
          {
               q=p;
               p=p->nextptr;
          }
          q->nextptr=stnode;
          printf("\n The deleted node is : %d\n",p->num);
          free(p);
}

void displayClList()
{
    struct node *tmp;
    int n = 1;

    if(stnode == NULL)
    {
        printf(" No data found in the List yet.");
    }
    else
    {
        tmp = stnode;
        printf("\n Data entered in the list are :\n");
        do
          {
            printf(" Data %d = %d\n", n, tmp->num);
            tmp = tmp->nextptr;
            n++;
        }while(tmp != stnode);
    }
}
```