```
/*********************SUBHAS NATH**********************/
/*********************************************************/
/******PROGRAM OF CIRCULAR LINKED LIST INSERTION******/

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct link
 {
  int info;
  struct link *next;
 };
typedef struct link node;
void main()
 {
  node *head1,*head2;
  clrscr();
  void create(node *,node *);
  void show(node *,node *);
  node *insert_fst(node *);
  node *insert_lst(node *);
  node *del_fst(node *);
  node *del_lst(node *);
  head1=(node *)malloc(sizeof(node)); //CREATE STORAGE FOR FIRST NODE
  head2=head1;
  printf("\n\tCreate the linked list:\n");
  create(head1,head2);
  printf("\n\tDisplay the linked list:\n");
  show(head1,head2);
  printf("\n\n\tInsertion at 1st position.\n");
  head1=insert_fst(head1);
  //head2=head1;
  show(head1,head1);
  printf("\n\n\tInsertion at last position.\n");
  head1=insert_lst(head1);
  show(head1,head1);
  getch();
  printf("\n\n\tDeletion at 1st position.\n");
  head1=del_fst(head1);
  show(head1,head1);
  getch();
  printf("\n\n\tDeletion at last position.\n");
  head1=del_lst(head1);
  show(head1,head1);
  getch();
 }
```

```
/******************CREATE LINKED LIST*******************/

void create(node *temp1,node *temp2)
 {
  char ans;
  printf("\nEnter the info:-   ");
  scanf("%d",&temp1->info);
  temp1->next=temp2;
  printf("\nWant Another?  ");
  fflush(stdin); //CLEAR THE BUFFER
  ans=getchar();
  if(ans!='y')
   return;
  else
   {
    temp1->next=(node *)malloc(sizeof(node));//CREATE STORAGE FOR NEXT
NODE
    create(temp1->next,temp2);  //RECURSIVE CALL TO CREATE NEXT NODE
   }
 }

/*******************DISPLAY LINKED LIST******************/

void show(node *temp1,node *temp2)
 {
  if(temp1->next==temp2)
   printf("%d",temp1->info); //DISPLAY LAST NODE
  else
   {
    printf("%d->",temp1->info);//DISPLAY CURRENT INFO
    show(temp1->next,temp2);   //RECURSIVE CALL TO DISPLAY NEXT INFO
   }
 }


 /*****************INSERTION AT FIRST POSITION**************/

node *insert_fst(node * temp)
 {
  node * list,*first;
  first=temp;
  list=(node *)malloc(sizeof(node));
  printf("\nEnter the element:-  ");
   scanf("%d",&list->info);
  list->next=temp;
  while(temp->next!=first)
   {
    temp=temp->next;
   }
  if(temp->next==first)
   {
    temp->next=list;
   }
  temp=list;
  return(temp);
 }



/******************INSERTION AT LAST POSITION*****************/

node *insert_lst(node *temp)
 {
  node *last,*first;
  first=temp;
  while(temp->next!=first)
   {
    temp=temp->next;
   }
  if(temp->next==first)
   {
    last=(node *)malloc(sizeof(node));
    printf("\nEnter the element:-  ");
     scanf("%d",&last->info);
    temp->next=last;
    last->next=first;
   }
  return(first);
 }


/****************DELETION FROM FIRST POSITION***************/

node *del_fst(node * temp)
 {
  node * list,*first;
  first=temp;
  temp=temp->next;
  list=temp;
  while(temp->next!=first)
    {
     temp=temp->next;
    }
  if(temp->next==first)
    {
     temp->next=list;
    }
  free(first);
  return(list);
 }



/***************DELETION FROM LAST POSITION***************/

node *del_lst(node *temp)
 {
  node *last,*first;
  first=temp;
  while(temp->next->next!=first)
   {
    temp=temp->next;
   }
  if(temp->next->next==first)
   {
    last=temp->next;
    free(last);
    temp->next=first;
   }
  return(first);
 }
```