Circular Linked List:

```
#include<stdio.h>
#include<stdlib.h>
struct node
{
   int data;
   struct node *next;
};
struct node *start;

void beginsert ();
void lastinsert ();
void begin_delete();
void last_delete();
void display();

void main ()
{
   int choice =0;
   while(choice != 7)
   {
      printf("\n*********Main Menu*********\n");
      printf("\nChoose one option from the following list ...\n");
      printf("\n===============================================\n");
      printf("\n1.Insert in begining\n2.Insert at last\n3.Delete from Beginning\n4.Delete from
last\n5.Show\n6.Exit\n");
      printf("\nEnter your choice?\n");
      scanf("\n%d",&choice);
      switch(choice)
      {
```

```
            case 1:
            beginsert();
            break;
            case 2:
            lastinsert();
            break;
            case 3:
            begin_delete();
            break;
            case 4:
            last_delete();
            break;
            case 5:
                        display();
            break;
            case 6:
            exit(0);
            break;
            default:
            printf("Please enter valid choice..");
        }
    }
}
void beginsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
```

```
    }
    else
    {
        printf("\nEnter the node data?");
        scanf("%d",&item);
        ptr -> data = item;
        if(start == NULL)
        {
            start = ptr;
            ptr -> next = start;
        }
        else
        {
            temp = start;
            while(temp->next != start)
                temp = temp->next;
            ptr->next = start;
            temp -> next = ptr;
            start = ptr;
        }
        printf("\nnode inserted\n");
    }

}
void lastinsert()
{
    struct node *ptr,*temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
```

```
        printf("\nOVERFLOW\n");
    }
    else
    {
        printf("\nEnter Data?");
        scanf("%d",&item);
        ptr->data = item;
        if(start == NULL)
        {
            start = ptr;
            ptr -> next = start;
        }
        else
        {
            temp = start;
            while(temp -> next != start)
            {
                temp = temp -> next;
            }
            temp -> next = ptr;
            ptr -> next = start;
        }

        printf("\nnode inserted\n");
    }

}

void begin_delete()
{
    struct node *ptr;
```

```
    if(start == NULL)
    {
        printf("\nUNDERFLOW");
    }
    else if(start->next == start)
    {
        start = NULL;
        free(start);
        printf("\nnode deleted\n");
    }

    else
    {   ptr = start;
        while(ptr -> next != start)
            ptr = ptr -> next;
        ptr->next = start->next;
        free(start);
        start = ptr->next;
        printf("\nnode deleted\n");

    }
}
void last_delete()
{
    struct node *ptr, *preptr;
    if(start==NULL)
    {
        printf("\nUNDERFLOW");
    }
    else if (start ->next == start)
    {
```

```
        start = NULL;
        free(start);
        printf("\nnode deleted\n");

    }
    else
    {
        ptr = start;
        while(ptr ->next != start)
        {
            preptr=ptr;
            ptr = ptr->next;
        }
        preptr->next = ptr -> next;
        free(ptr);
        printf("\nnode deleted\n");

    }
}



void display()
{
    struct node *ptr;
    ptr=start;
    if(start == NULL)
    {
        printf("\nnothing to print");
    }
    else
```

```
    {
        printf("\n printing values ... \n");

        while(ptr -> next != start)
        {

            printf("%d\n", ptr -> data);
            ptr = ptr -> next;
        }
        printf("%d\n", ptr -> data);
    }

}
```