```c
/*
Title- Circular Queue Implimentation using array(statically)
Author- Bhakare Mahesh Santosh
ID- 492
Batch- TechnOrbit(PPA-8)
*/

#include<stdio.h>
#define MAX 10

//  ---------------- STRUCTURE DECLARATION -----------------
struct QUEUE
{
    int arr[MAX];
    int front;
    int rear;
    int count;
};

// ------------------- QUEUE INITILIZATION -------------------

void init_queue(struct QUEUE* queue)
{
    queue->front = -1;
    queue->rear = -1;
    queue->count = 0;
}

//  --------------------- TO CHECK QUEUE IS FULL OR NOT ----------------------

int full(struct QUEUE* queue)
{
    if(queue->count == MAX)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

//  ------------------------------ TO CHECK QUEUE IS EMPTY OR NOT ----------------------

int empty(struct QUEUE* queue)
{
    if(queue->count == 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}


//  ------------------- FUNCTION TO INSERT ELEMENT IN QUEUE ----------------

void enqueue(struct QUEUE* queue)
{
    queue->rear = ((queue->rear)+1) % MAX;
    printf("Enter data in Queue: ");
    scanf("%d",&(queue->arr[queue->rear]));
```

```
    (queue->count)++;
}

// --------------------- FUNCTION TO DELETE ELEMENT FROM QUEUE ----------------

void dequeue(struct QUEUE* queue)
{
    queue->front = ((queue->front)+1)%MAX;
    printf("Popped element is: %d\n",queue->arr[queue->front]);
    (queue->count)--;
}


// ---------------------------- FUNCTUON TO DELETE QUEUE --------------------

void display(struct QUEUE* queue)
{
    printf("Your Queue is: ");
    int i;
    for(i = ((queue->front)+1)%MAX ; i != ((queue->rear))%MAX ; i = (i+1)%MAX)
    {
        printf("%d <= ",queue->arr[i]);
    }
    printf("%d <= ",queue->arr[i]);
}


// ------------------------------------ ENTRY POINT FUNCTION --------------------------

void main()
{
    int choice;
    struct QUEUE queue;
    init_queue(&queue);
    do
    {
        printf("**********************************\n");
        printf("1. Enqueue\n2. Dequeue\n3. IsFull\n4. IsEmpty\n5. Display\n6. Exit\nEnter
Your Choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: if(full(&queue))
                    {
                        printf("Queue is Full.........\n");
                    }
                    else
                    {
                        enqueue(&queue);
                    }
                    break;

            case 2: if(empty(&queue))
                    {
                        printf("Queue is Empty.....\n");
                    }
                    else
                    {
                        dequeue(&queue);
                    }
                    break;

            case 3: if(full(&queue))
                    {
                        printf("Queue is full....\n");
                    }
```

```
                    else
                    {
                        printf("Queue is not Full...\n");
                    }
                    break;
            case 4: if(empty(&queue))
                    {
                        printf("Queue is empty....\n");
                    }
                    else
                    {
                        printf("Queue is not Empty....\n");
                    }
                    break;

            case 5: if(empty(&queue))
                    {
                        printf("Queue is Empty...\n");
                    }
                    else
                    {
                        display(&queue);
                    }
                    break;

            case 6: printf("Exiting.........\n");
                    break;

            default : printf("Wrong choice please enter proper choice.........\n");
        }

    }while(choice != 6);

}
```