```
//Created By Ritwik Chandra Pandey on 24/02/21
//183215
//Circular Queue Using Linked List

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

struct queue {
    int data;
    struct queue *next;
};

typedef struct queue *CircularQueue;
CircularQueue front = NULL, rear = NULL;

void enqueue(int element) {
    CircularQueue temp = NULL;
    temp = (CircularQueue)malloc(sizeof(struct queue));
    if(temp == NULL) {
        printf("Circular queue is overflow.\n");
    } else {
        temp -> data = element;
        temp -> next = NULL;
        if(front == NULL) {
            front = temp;
        } else {
            rear -> next = temp;
        }
        rear = temp;
        rear -> next = front;
        printf("Successfully inserted.\n");
    }
}

void dequeue() {
    CircularQueue temp = NULL;
    if(front == NULL) {
        printf("Circular queue is underflow.\n");
    } else {
        temp = front;
        if (front == rear) {
            front = rear = NULL;
        } else {
            front = front -> next;
            rear -> next = front;
        }
        printf("Deleted value = %d\n", temp -> data);
        free(temp);
    }
}

void display() {
    if(front == NULL) {
        printf("Circular queue is empty.");
    } else {
        CircularQueue temp = front;
        printf("Elements are : ");
        do {
            printf("%d ", temp -> data);
            temp = temp -> next;
        } while(temp != front);
    }
    printf("\n");
}


CircularQueue deleteList(CircularQueue head_ref)
{
    CircularQueue current = head_ref;
    CircularQueue next;
    do{


        next = current->next;
        free(current);
        current = next;
    }while(current!=head_ref);
    head_ref = NULL;
    return head_ref; }


int main(){

    int selection=0,x;
    printf("CIRCULAR QUEUE LINKED LIST IMPLEMENTATION\n\n");

           do{



               printf("\t1.ENQUEUE\n\t2.DEQUEUE\n\t3.DISPLAY\n\t4.DELETE CICULAR QUEUE\n\t5.EXIT\n");

               printf("\t\n Please enter your choice\n");
               scanf("%d",&selection);


               switch(selection)
               {

                   case 1:

                       printf("Enter Value\n");
                       scanf("%d",&x);
                       enqueue(x);
                       break;

                   case 2:
                       dequeue();
                       break;
                   case 3:
```

```
            display();
            break;
        case 4:
            front=deleteList(front);
            rear=NULL;
            printf("Circular Queue has been deleted\n");
            break;
        case 5:
            exit(1);
            break;
        default:
            printf("\t\n\nYou have not entered the right choice\n\n");
    }}
while(selection!=5);

}
```