main.c

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
int front=-1;
int rear=-1;
int queue[MAX];
void Enque(int);
int Deque();
void display();
int main(int argc, char **argv){
    int option;
    int item;
    do{
        printf("\nCircular Queue\n");
        printf("\n 1. Insert to Queue (EnQueue)");
        printf("\n 2. delete from the Queue (DeQueue)");
        printf("\n 3. Display the content ");
        printf("\n 4. Exit\n");
        printf("Enter the option :");
        scanf("%d",&option);
        switch(option){
            case 1:  printf("Enter the element\n");
                     scanf("%d",&item);
                     Enque(item);
                     break;
            case 2: item=Deque();
                    if(item==-999)
                        printf("Queue is empty");
                    else
                    printf("Removed element from the queue %d",item);
                    break;
            case 3: display();
                    break;
            case 4: exit(0);
        }
    } while (option!=4);
```

```
    } while (option!=4);
    return 0;
}

void Enque(int ele){
    if(((front == 0 && rear == MAX - 1))|| (front == rear + 1) ){
        printf("Queue is full\n");
        return;
    }
    else{
      rear=(rear+1)%MAX;
      queue[rear]=ele;
      if(front ==-1)
          front=0;
    }
}
int Deque(){
    int item;
    if((front == -1)&&(rear == -1)){
        return(-999);
    }
    else {
        item=queue[front];
        if(front==rear){
            front=-1;
            rear=-1;
        }
        else{
            front=(front+1)%MAX;
        }
        return item;
    }
}
void display(){
    int i;
    if((front==-1) && (rear==-1))|| (front==rear)){
```

```
    }
    else{
      rear=(rear+1)%MAX;
      queue[rear]=ele;
      if(front ==-1)
          front=0;
    }
}
int Deque(){
    int item;
    if((front == -1)&&(rear == -1)){
        return(-999);
    }
    else {
        item=queue[front];
        if(front==rear){
            front=-1;
            rear=-1;
        }
        else{
            front=(front+1)%MAX;
        }
        return item;
    }
}
void display(){
    int i;
    if(((front==-1)&& (rear==-1))|| (front==rear)){
        printf("Queue is empty\n");return;
    }
    else{
        printf("\n Queue contents:\n");
        for(i=front;i<=rear;i++)
            printf("%d\t", queue[i]);
    }
}
```

input

```
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
4

Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
5

Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
6
Queue is full

Circular Queue

 1. Insert to Queue (EnQueue)
```

```
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
1

Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
2

Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :1
Enter the element
3

 Circular Queue

  1. Insert to Queue (EnQueue)
```

```
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :2
Removed element from the queue 1
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :2
Removed element from the queue 2
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :3

 Queue contents:
3       4       5
Circular Queue

 1. Insert to Queue (EnQueue)
 2. delete from the Queue (DeQueue)
 3. Display the content
 4. Exit
Enter the option :
```