Source Code -

```
/*
 Author - Neevan Redkar
 Class - SE
 Branch - IT
 Roll no - 3069
 Batch S3
*/
#include <iostream>
using namespace std;

int enqueue(string[],int,int,int);
int dequeue(string[],int,int,int);
void print(string[],int,int,int);
string isFull(int,int,int);
string isEmpty(int,int);

int main(){
    int length,front,rear;
    cout<<"Enter length of queue\n";
    cin>>length;
    string queue[length],repeat="";
    front=-1;
    rear=-1;
    do{
        cout<<"Enter operation of choice\n1.Enqueue\n2.Dequeue\n3.Print\n4.Check if
full\n5.Check if empty"<<endl;
        int choice;
        cin>>choice;
        switch(choice){
            case 1:
            rear = enqueue(queue,length,front,rear);
            if(front==-1&&rear!=-1){
                front=0;
            }
            break;
            case 2:
            front=dequeue(queue,front,rear,length);
            if(front==-1&&rear!=-1){
                rear=-1;
            }
            break;
            case 3:
            print(queue,front,rear,length);
            break;
            case 4:
            cout<<isFull(front,rear,length)<<endl;
            break;
            case 5:
            cout<<isEmpty(front,rear)<<endl;
            break;
            default:
            cout<<"Invalid operation"<<endl;
        }
        cout<<"Do you wish to continue (yes/no)(case sensitive)"<<endl;
        cin>>repeat;
    }while(repeat == "yes");


    return 0;
}

int enqueue(string queue[],int length,int front,int rear){

    if (rear!=-1&&(rear+1)%length==front){
        cout<<"Queue is full"<<endl;
```

```
        print(queue,front,rear,length);
        return rear;
    }else if(rear==-1&&front==-1){
        rear=0;front=0;
        string element;
        cout<<"Enter element"<<endl;
        cin>>element;

        queue[rear]=element;
        cout<<"Queue is now \n";

        print(queue,front,rear,length);
        return rear;

    }else{

        string element;
        cout<<"Enter element"<<endl;
        cin>>element;
        rear=(rear+1)%length;
        queue[rear]=element;

        cout<<"Queue is now \n";
        print(queue,front,rear,length);

     return rear;
    }

}

int dequeue(string queue[],int front,int rear,int length){
    if(front==-1){
        cout<<"Queue is empty"<<endl;
        return front;
    }else if(front==rear){
        cout<<"Dequeued element is "<<queue[front];
        cout<<"\nQueue is now empty"<<endl;
        front=-1;
        return front;
    }else{
        cout<<"Dequeued element is "<<queue[front]<<endl;
        front=(front+1)%length;

        print(queue,front,rear,length);
        return front;
    }

}

string isFull(int front,int rear,int length){
    if (rear!=-1&&(rear+1)%length==front){
        return "Queue is full";
    }else{
        return "Queue not full";
    }

}

void print(string queue[],int front,int rear,int length){
    cout<<"[ ";
    if(rear<front){
        int l=length+rear;
        for(int i=front;i<=l;i++){
             cout<<queue[(i%length)]<<",";
        }
```

```
    }else{
        for(int i=front;i<=rear;i++){
            cout<<queue[(i%length)]<<",";
        }
    }
    cout<<"]";
    cout<<endl;
}

string isEmpty(int front,int rear){
    if (rear==-1&&front==-1){
        return "Queue is empty";
    }else{
        return "Queue not empty";
    }

}
```

Output -

```
pvg@pvg-HP-ProDesk-400-G4-SFF:~/Desktop/SE_IT_3069/Assignments/Assignment3$ ./-
CircularQueue
Enter length of queue
5
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
5
Queue is empty
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Car
Queue is now
[ Car,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
5
Queue not empty
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
```

```
Enter element
Bus
Queue is now
[ Car,Bus,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Bike
Queue is now
[ Car,Bus,Bike,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Train
Queue is now
[ Car,Bus,Bike,Train,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Aeroplane
Queue is now
[ Car,Bus,Bike,Train,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
4
Queue is full
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Queue is full
[ Car,Bus,Bike,Train,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
```

```
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Car
[ Bus,Bike,Train,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
4
Queue not full
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Bus
[ Bike,Train,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Bike
[ Train,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Train
[ Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Aeroplane
Queue is now empty
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
```

```
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Truck
Queue is now
[ Truck,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Tram
Queue is now
[ Truck,Tram,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
2
Dequeued element is Truck
[ Tram,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
3
[ Tram,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Truck
Queue is now
[ Tram,Truck,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
```

```
Enter element
Car
Queue is now
[ Tram,Truck,Car,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Bus
Queue is now
[ Tram,Truck,Car,Bus,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
1
Enter element
Aeroplane
Queue is now
[ Tram,Truck,Car,Bus,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
yes
Enter operation of choice
1.Enqueue
2.Dequeue
3.Print
4.Check if full
5.Check if empty
3
[ Tram,Truck,Car,Bus,Aeroplane,]
Do you wish to continue (yes/no)(case sensitive)
no
```