

```

#include<iostream>
using namespace std;

class CircularQueue{

private:
int *queue;
int rear,front,size;
public :
CircularQueue(int s){

    size=s;
    queue=new int[size];
    front=-1;
    rear=-1;
}
~ CircularQueue(){
    delete[]queue;
}
void enqueue(int value){
    if ((rear+1)%size==front){
        cout<<"Queue is full cannot enqueue"<<value<<endl;

    }else{
        if(front== -1){
            front=0;
        }
        rear=(rear+1)%size;
        queue[rear]=value;
        cout<<"The value is enqueued"<<value<<endl;

    }
}

void dequeue(){
    if(front== -1){
        cout<<"queue is empty"<<endl;

    }else{
        cout<<"dequed"<<queue[front]<<endl;
        if(front==rear){
            front=-1;
            rear=-1;

        }else{
            front=(front+1)%size;

        }

    }
}

void display(){
    if(front== -1){
        cout<<"queue is empty"<<endl;
    }
}

```

```

        }else{
            cout<<"queue elements";
            int i=front;
            while(true){
                cout<<queue[i]<<" ";
                if(i==rear){
                    break;
                }
                i=(i+1)%size;
            }
            cout<<endl;
        }
    }
};

int main(){
    int size;
    cout<<"Enter the sizeof circular queue:\n"<< endl;
    cin>>size;
    CircularQueue cq(size);

    int choice,value;
    do{
        cout<<"Circular queue operation ::";
        cout<<"1.Enqueue\n2.Dequeue\n3.Display\n4.Exist\n";
        cout<<"Enter the choice: ";
        cin>>choice;

        switch(choice){
            case 1:
                cout<<"enter the value of enqueue:";
                cin>>value;
                cq.enqueue(value);
                break;
            case 2:
                cq.dequeue();
                break;
            case 3:
                cq.display();
                break;
            case 4:
                cout<<"Existing.....";
                break;

            default:
                cout<<"Invalid Choice. try again"<<endl;
        }
    }while(choice!=4);
    return 0;
}

```