```
class MyQueue:
    def __init__(self):
        self.queue = []
    def enqueue(self, item):
        self.queue.append(item)
    def dequeue(self):
        if len(self.queue)==0:
            print("no element to pop")
        else:
            return self.queue.pop()
    def is_empty(self):
        return len(self.queue) == 0

    def display(self):
        if self.queue:
            return self.queue
        else:
            return "NO QUEUE TO DISPLAY"
    def size (self):
        return len(self.queue)

queue = MyQueue()
queue.enqueue(10)
queue.enqueue(20)
queue.enqueue(30)
queue.display()
```

```
[10, 20, 30]
```

```
queue.dequeue()
queue.display()
```

```
[10, 20]
```

```
queue.dequeue()
queue.display()
```

```
[10]
```

```
queue.dequeue()
queue.display()
```

```
'NO QUEUE TO DISPLAY'
```

```
queue.is_empty()
```

```
True
```

```
queue.size()
```

```
0
```

CIRCULAR QUEUE WITH FIXED SIZE

```
class circularQ:
    def __init__(self,size):
        self.size = size
        self.front = self.rear = -1
        self.queue = [None]*size
    def enqueue(self,item):
        if (self.rear+1) %self.size == self.front:
            return "QUEUE IS FULL!!!!!"
        elif self.front == -1:
            self.front = self.rear = 0
            self.queue[self.rear] = item
        else:
            self.rear = (self.rear+1)% self.size
            self.queue[self.rear] = item
    def dequeue(self):
        if self.front == -1:
            return "THE QUEUE IS EMPTY"
        elif self.front == self.rear:
            popped = self.queue [self.front]
            self.front = self.rear = -1
            return popped
        else:
            popped = self.queue[self.front]
            self.front = (self.front +1)%self.size
            return popped
    def display(self):
        if self.front == -1:
            return "QUEUE IS EMPTY"
        i = self.front
        while True:
            print(self.queue[i])
            if i == self.rear:
                break
            i = (i+1) % self.size

Q=circularQ(4)
Q.enqueue(98)
Q.enqueue(98)
Q.enqueue(87)
Q.enqueue(88)
Q.enqueue(5)
```

```
'QUEUE IS FULL!!!!!'
```

```
Q.dequeue()
Q.dequeue()
Q.dequeue()
```

```
Q.dequeue()
Q.dequeue()
```

```
'THE QUEUE IS EMPTY'
```

```
q=circularQ(3)
q.enqueue(98)
q.enqueue(98)
q.enqueue(87)
q.display()
```

```
98
98
87
```

```
q=circularQ(3)
q.display()
```

```
'QUEUE IS EMPTY'
```