create a class called 'MyQueue' which can handle 1.Enqueue 2.Dequeue 3.Is empty 4.Display()

```
class Myqueue:
    def __init__(self):
        self.queue=[]
    def enqueue(self,data):
        self.queue.append(data)
    def dequeue(self):
        if self.queue:
            self.queue.pop(0)
        else:
            print("queue is empty")
    def isempty(self):
        return len(self.queue)==0
    def display(self):
        if self.queue:
            return self.queue
        else:
            print('no queue to display')
    def size(self):
        return len(self.queue)





j = Myqueue()
j.enqueue(90)
j.enqueue(85)
j.enqueue(60)
j.enqueue(40)
j.display()
```

```
[90, 85, 60, 40]
```

```
j.dequeue()
j.display()
```

```
[85, 60, 40]
```

```
j.size()
```

```
3
```

write a function that accepts an array arr of size 'length' as its argument you are reqired to

```
arr=[5,7,8,3,2,6]
e = []
o = []
```

```
for i in range(len(arr)):
    if i%2 == 0:
        e.append(arr[i])
    else:
        o.append(arr[i])
print(e)
print(o)
e.sort(reverse = True)
print("sorted  even array:",e)
print(e[0])
o.sort(reverse = True)
print("sorted odd array:",o)
print(o[0])
```

```
[5, 8, 2]
[7, 3, 6]
sorted  even array: [8, 5, 2]
8
sorted odd array: [7, 6, 3]
7
```

```
def sum_e_o(arr):
    even = []
    odd = []
    for i in range(len(arr)):
        if i%2 == 0:
            even.append(arr[i])
        else:
            odd.append(arr[i])
    even.sort(reverse=True)
    odd.sort()
    return e[1]+o[1]
```

```
print(sum_e_o(arr))
```

```
11
```

```
class circularQ:
    def __init__(self,size):
        self.size = size
        self.front=self.rear=-1
        self.queue=[None]*size
    def enqueue(self,data):
```