```
class Queue:


    def init(self, max_size):
        self.max_size = max_size
        self.Q = [0] * max_size
        self.num = 0
        self.first = 0
    def enqueue(self, item):
        if self.num >= self.max_size:
            raise Exception("Queue overflow")
        self.Q[(self.num + self.first) % self.max_size] = item
        self.num += 1
    def dequeue(self):
        if self.num == 0:
            raise Exception("Queue empty")
        item = self.Q[self.first]
        self.first = (self.first + 1) % self.max_size
        self.num -= 1
        return item
    def front(self):
        if self.num == 0:
            raise Exception("Queue empty")
        return self.Q[self.first]
    def is_empty(self):
        return self.num == 0
    def size(self):
```

```
    return self.num
def is_full(self):
    return self.num >= self.max_size

def dequeue_element(self , element):
    if self.num == 0:
        raise Exception("Queue empty")
    index_list=[]
    for i in range(self.num):
        if self.Q[(self.first + i) % self.max_size] == element:
            index_list.append((self.first + i) % self.max_size)
    for j in index_list:
        self.Q[j]=None
    for k in self.Q:
        if self.Q[k]==None:
            del self.Q[k]
            index=[(self.num + self.first) % self.max_size]
            self.Q.insert(0,index)
            self.num-=1
    print(self.Q)
    return self.Q

def dequeue_i(self , i):
    if self.num == 0:
        raise Exception("Queue empty")
    item=self.Q[i]
```

```
        del self.Q[i]
        index=[(self.num + self.first) % self.max_size]
        self.Q.insert(0,index)
        self.num-=1
        return item



#Example

q=Queue(10) # (front of queue)[](back of queue)
q.enqueue("ra'na") # ["ra'na"]
q.enqueue("vez") # ["ra'na", "vez"]
q.enqueue("Arya") # ["ra'na", "vez", "Arya"]
print("queue size is: ",q.size())
print(q.dequeue(), "left the queue") # ["vez", "Arya"]
print("front of queue is:",q.front())
q.enqueue("milda") # ["vez", "Arya", "milda"]
q.dequeue() # ["Arya","milda"]
q.dequeue() # ["milda"]
q.dequeue() # []

print("It was a queue")
```