

ΑΣΚΗΣΗ 1η

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί μία ουρά με τη χρήση ενός πίνακα.

```
class Queue:
    # Constructor
    def __init__(self):
        self.queue = list()
        self.maxSize = 8
        self.head = 0
        self.tail = 0

    def is_empty(self):
        return len(self.queue) == 0

    # Adding elements
    def enqueue(self, data):
        # Checking if the queue is full
        if self.size() >= self.maxSize:
            return ("Queue is Full")
        self.queue.append(data)
        self.tail += 1
        return True

    # Deleting elements
    def dequeue(self):
        # Checking if the queue is empty
        if self.size() <= 0:
            self.resetQueue()
            return ("Queue is Empty")
        data = self.queue[self.head]
        self.head += 1
        return data

    # Calculate size
    def size(self):
        return self.tail - self.head

    # Reset queue
    def resetQueue(self):
        self.tail = 0
        self.head = 0
        self.queue = list()

myQueue = Queue()
number = int(input('Next item:'))
while number != 0:
    Queue.enqueue(myQueue, number)
    print(myQueue.queue)
    number = int(input('Next item:'))

print('The elements of the Queue are', myQueue.queue)

while not Queue.is_empty(myQueue):
    number = Queue.dequeue(myQueue)
    print('Remove from the queue the item:', number)
```

ΑΣΚΗΣΗ 2η

Να μελετηθεί και να εκτελεστεί ο παρακάτω κώδικας που υλοποιεί μία ουρά με τη χρήση μίας λίστας.

```
class Queue:
    # Constructor creates a list
    def __init__(self):
        self.queue = list()

    def is_empty(self):
        return len(self.queue) == 0

    # Adding elements to queue
    def enqueue(self,data):
        # Checking to avoid duplicate entry (not mandatory)
        if data not in self.queue:
            self.queue.insert(0,data)
            return True
        return False

    # Removing the last element from the queue
    def dequeue(self):
        if len(self.queue) > 0:
            return self.queue.pop()
        return ("Queue Empty!")

    # Getting the size of the queue
    def size(self):
        return len(self.queue)

myQueue = Queue()
number = int(input('Next item:'))
while number != 0:
    Queue.enqueue(myQueue,number)
    print(myQueue.queue)
    number = int(input('Next item:'))
print(myQueue.queue)
while not Queue.is_empty(myQueue):
    number = Queue.dequeue(myQueue)
    print(number)
    print(myQueue.queue)
    print(myQueue.dequeue())
```