```
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class Queue:
    def __init__(self):
        self.front = None
        self.rear = None

    def is_empty(self):
        return self.front is None

    def enqueue(self, data):
        new_node = Node(data)
        if self.rear is None:
            # Queue is empty
            self.front = self.rear = new_node
            return
        self.rear.next = new_node
        self.rear = new_node

    def dequeue(self):
        if self.is_empty():
            print("Queue is empty")
            return None
        removed_data = self.front.data
        self.front = self.front.next

        if self.front is None:
            # Queue became empty
            self.rear = None

        return removed_data

    def peek(self):
        if self.is_empty():
            print("Queue is empty")
            return None
        return self.front.data

    def display(self):
        if self.is_empty():
            print("Queue is empty")
            return
        current = self.front
        while current:
            print(current.data, end=" -> ")
            current = current.next
        print("None")
q = Queue()
```

```
q.enqueue(10)
q.enqueue(20)
q.enqueue(30)

q.display()        # 10 -> 20 -> 30 -> None
print(q.dequeue()) # 10
q.display()        # 20 -> 30 -> None
print(q.peek())    # 20
```