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


class Queue:
    def __init__(self):
        self.front = 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:
            self.front = self.rear = new_node
            return
        self.rear.next = new_node
        self.rear = new_node

    def dequeue(self):
        if self.is_empty():
            return None
        data = self.front.data
        self.front = self.front.next
        if self.front is None:
            self.rear = None
        return data
```

```
    def peek(self):
        if self.is_empty():
            return None
        return self.front.data

    def display(self):
        current = self.front
        while current:
            print(current.data, end=" ")
            current = current.next
        print()

queue = Queue()
n = int(input())
for i in range(n):
    data= int(input())
    queue.enqueue(data)

print(f"Front of the queue: {queue.peek()}")

print("Queue elements:")
queue.display()

print("Dequeue elements:")
while not queue.is_empty():
    print(queue.dequeue())
```