Read over the following solution for the function `list_queue` you worked on during the lab. We have omitted most of the docstring, and have added two `print(q)` statements.

```python
def list_queue(list_, q):
    """
    :param list_: a Python list, possibly noted
    :type list_ : list
    :param q: an empty queue
    :type q : Queue
    :rtype: None
    """
    for i in list_:
        q.add(i)

    print(q)
    while not q.is_empty():
        el = q.remove()
        if isinstance(el, list):
            for j in el:
                q.add(j)
        else:
            print(el)

    print(q)
```

Remember that the `Queue` class in lab has three methods (other than `__init__`):

- `add(o)`: add object `o` at the end of this queue.
- `remove()`: remove and return object at beginning of queue.
- `is_empty()`: Return whether queue is empty or not.

Assume `print(q)` prints a user-friendly string representation of `Queue q`.

Assume that `L = ['a', 'b', 'c', 'd'], ['e', 'g'], 'f']` and that `q` is an empty `Queue`. Write the output of the function call `list_queue(L, q)`. (We have written the first line of the output for you.)
['a', ['b', ['c', 'd']], ['e', 'g'], 'f']
a
[['b', ['c', 'd']], ['e', 'g'], 'f']
[['e', 'g'], 'f', 'b', ['c', 'd']]
[['f', 'b', ['c', 'd']], 'e', 'g']
f
['b', ['c', 'd'], 'e', 'g']
b
[['c', 'd'], 'e', 'g']
[['e', 'g'], 'c', 'd']
e
['g', 'c', 'd']
g
['c', 'd']
c
['d']
d
[]