Deep Count

```
def deepcount(seq):
    if seq == []:
        return 0
    elif type(seq) != list:
        return 1
    else:
        return deepcount(seq[0]) + deepcount(seq[1:])
```

Deep Map

```
def deepMap(func,seq):
    if seq == []:
        return seq
    elif type(seq) != list
        return func(seq)
    else:
        return [deepSquare(func,seq[0])] + deepSquare(func,seq[1:])
```

Change list to tuple deeply

```
def deep_tuple(s):
    if not isinstance(s, list): return s
    return tuple(deep_tuple(i) for i in s)
```

Flatten

```
def flatten(seq):
    if seq == []:
        return seq
    elif type(seq) != list:
        return [seq]
    else:
        return flatten(seq[0]) + flatten(seq[1:])
```

Deep Count

```
def deepcount(seq):
    if seq == []:
        return 0
    elif type(seq) != list:
        return 1
    else:
        return deepcount(seq[0]) + deepcount(seq[1:])
```

Deep Map

```
def deepMap(func,seq):
    if seq == []:
        return seq
    elif type(seq) != list
        return func(seq)
    else:
        return [deepSquare(func,seq[0])] + deepSquare(func,seq[1:])
```

Change list to tuple deeply

```
def deep_tuple(s):
    if not isinstance(s, list): return s
    return tuple(deep_tuple(i) for i in s)
```

Flatten

```
def flatten(seq):
    if seq == []:
        return seq
    elif type(seq) != list:
        return [seq]
    else:
        return flatten(seq[0]) + flatten(seq[1:])
```