Challenge

Problem 1) Can you write a counter function (or function producer) without using objects and without using global variables?

Two hints: python supports HOFs and nested function declarations. To access an outer scoped variable, use the nonlocal keyword. Here is a convoluted “increment” function that shows some of the features you will need.

```python
def inc(x):
    i = x
    def doit():
        nonlocal i
        i = i + 1
        doit()
    return i
```

Example 1) Consider this Python code.

```python
class Delay:
    def __init__(self, action):
        self.action = action
        self.status = 0

    def report(x):
        print("Thunk executed: {}".format(x))
        return x

    def force(self):
        if self.status == 2:
            return self.value
        else:
            self.status = 1
            self.value = report(self.action())
            self.status = 2
            return self.value
```

• Review this code with a partner. Can you figure out how to use it?
• What is the purpose of `self.status = 1`?
Consider this function and list definition.

```python
def lazyTake(n,x):
    if x == {} or n < 1:
        return {}
    else:
        return {'head': x['head'],
                'tail': lazyTake(n-1, x['tail'].force())}
```

```python
l1 = {'head':2,
      'tail': Delay(lambda: {'head':3,
                             'tail': Delay(lambda: {'head':5,
                                                  'tail': Delay(lambda: {})})})}
```

**Problem 2)** Can you write `lazyTail`, `lazyMap`, and `lazyZipWith`?

**Problem 3)** Use these functions to make the infinite list of natural numbers and Fibonacci numbers.