

1. (5 points) LoopList

Implement a class LoopList that has a method at_index. If at_index is called with an index that is too large, the LoopList will loop around back to the beginning.

```
class LoopList:
    """
    >>> x = LoopList([3, 1, 4])
    >>> [x.at_index(i) for i in range(10)] # loops around!
    [3, 1, 4, 3, 1, 4, 3, 1, 4, 3]
    """
    def __init__(self, lst):
        self.lst = lst

    def at_index(self, i):
        return self.lst[i % len(self.lst)]
```

2. (5 points) Berkeley Landmark #158

Draw the environment diagram. (This is very hard!)

```
def campa(nile):
    def ding(ding):
        nonlocal nile
        def nile(ring):
            return ding
        return nile(ding(1914)) + nile(1917)
    return nile(ding(1914)) + nile(1917)
```

```
ring = campa(lambda nile: 100)
```

