CS SCHOLARS SEMINAR

February 28 and March 2, 2017

1. Write a function, `looper` that takes in a list of functions. It returns a function that takes in one argument, `x` and applies a function from `lst` to `x`. Please ignore the commented out line (it will be used in Problem 2).

```
def looper(lst):
    """
    >>> a, b, c = lambda x: x, lambda x: x * 2, lambda x: x + 1
    >>> actual_loop = looper([a, b, c])
    >>> actual_loop(1)
    1 # 1
    >>> actual_loop(2)
    4 # 2 * 2
    >>> actual_loop(3)
    4 # 3 + 1
    >>> actual_loop(3)
    3 # 3
    """
    def actual_loop(x):
        _____

        f = _____

        # keep_count(f)

        lst = _____

        return _____

    return actual_loop
```

2. **Challenge:** Now augment the code above so that we also keep track of how many times each function in `lst` was called. Assume that `actual_loop` behaves correctly. Now assume that `keep_count(f)` from the code above is uncommented. Fill in the function `keep_count` so that it also keeps track of how times each function was called. Write a function `count` that takes in a function and returns the number of times it was called.

```python
def looper(lst):
    """
    >>> a, b, c = lambda x: x, lambda x: x * 2, lambda x: x + 1
    >>> actual_loop, counter = looper([a, b, c])
    >>> counter(a)
    0
    >>> actual_loop(1)
    1
    >>> counter(a)
    1
    >>> actual_loop(2)
    4
    >>> actual_loop(3)
    4
    >>> actual_loop(3)
    3
    >>> counter(a)
    2
    """
    counts = _____

    def actual_loop(x):
        # assume correctly behaves as described in problem 1

    def keep_count(f):


        _____


    def count(f):


        _____


    return actual_loop, count
```