Nonlocal: exam-level questions

> If you need help reviewing Nonlocal, take a look at these resources:
>
> - Albert's and Robert's lectures (https://docs.google.com/presentation/d/1PC5Yw-AxxOyTaPhZ-kJ0JvVMwaVCyVmBmWdiNqGZeVo/edit#slide=id.g5b71800b6_1_94)

Each question has a "Toggle Solution" button -- click it to reveal that question's solution.

Code-Writing questions

Question 1

Implement a function `make_sassy_function` which takes a function `f` and returns a modified version of `f`: the new function should only work every other function call. The other half of the time, it should return a rude message.

```
def make_sassy_function(f, msg):
    """Returns a version of f that only works every other function
    call.

    >>> f = lambda x: x**2
    >>> sassy_f = make_sassy_function(f, 'Um, excuse me?')
    >>> sassy_f(4)
    16
    >>> sassy_f(5)
    'Um, excuse me?'
    >>> sassy_f(6)
    36
    >>> g = lambda x, y: x*y
    >>> sassy_g = make_sassy_function(g, "Ain't nobody got time for that!")
    >>> sassy_g(1, 2)
    2
    >>> sassy_g(5, 4)
    "Ain't nobody got time for that!"
    """
```

Toggle Solution

Question 2

Implement a function `sentence_buffer` which returns another one-argument function. This function will take in a word at a time, and it saves all the words that it has seen so far. If takes in a word that ends in a period ("."), that denotes the end of a sentence, and the function returns all the words in the sentence. It will also clear its memory, so that it no longer remembers any words.

```
def sentence_buffer():
    """Returns a function that will return entire sentences when it
    receives a string that ends in a period.

    >>> buffer = sentence_buffer()
    >>> buffer("This")
    >>> buffer("is")
    >>> buffer("Spot.")
    'This is Spot.'
    >>> buffer("See")
    >>> buffer("Spot")
    >>> buffer("run.")
    'See Spot run.'
    """
    "*** YOUR CODE HERE ***"
```

Toggle Solution