6.101 Recitation 14: Week 8 Mines Midpoint **4/3/24**

This sheet is yours to keep!

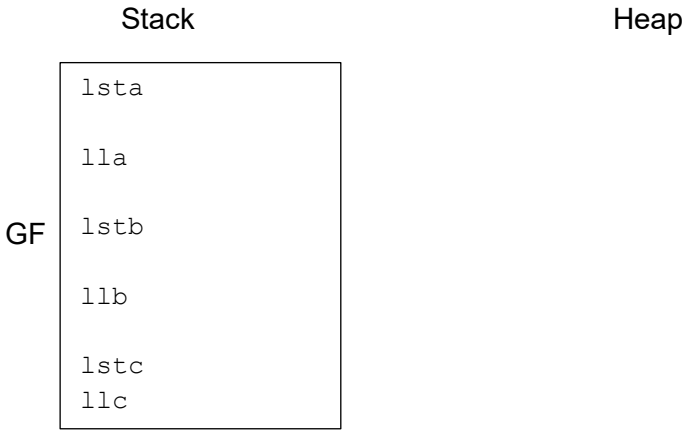
Question 1: For today's recitation we will define an empty linked list as None, and a non-empty linked list as a length two tuple of `(some_element, linked_list)`.

Complete the environment diagram to represent the execution of the code below.

```
lsta = [1, 2, 3]
lla = (1, (2, (3, None)))

lstb = [3]
llb = (3, None)

lstc = []
llc = None
```



Question 2: Fill in the body of the functions below:

```
def first(ll):
    """
    returns the first element of a non-empty linked list

    >>> first( (5, (10, (15, None))) )
    5
    """




def rest(ll):
    """
    returns the rest of a nonempty linked list
    (omitting the first element)

    >>> rest( (5, (10, (15, None))) )
    (10, (15, None))
    """
```

Question 3: Fill in the body of the functions below:

```
def ll_len(ll):
    """
    get the length of a linked list

    >>> ll_len( ('a',('b',None)) )
    2
    """
```

```
def ll_get(ll, i):
    """
    get the ith element of a linked list

    >>> ll_get( ('a',('b',None)), 1)
    'b'
    """
```

```
def make_ll(*elements):
    """
    given an arbitrary number of elements as arguments,
    make a linked-list of (first,rest) pairs

    >>> make_ll(1,2,3)
    (1, (2, (3, None)))
    """
```

R14 Participation Credit **Kerberos : _____@mit.edu**

Hand this sheet in at the end of recitation to get participation credit for today.

Question 3: Fill in the body of the function below:

```
def ll_elements(ll):
    """
    return a generator that yields each element in a linked list
    >>> ll_gen = ll_elements(make_ll(1, 2, 3))
    >>> next(ll_gen)
    1
    >>> list(ll_gen)
    [2, 3]
    """
```

Question 4: Fill in the body of the functions below:

```
def ll_plus(ll1, ll2):
    """
    return a new linked list that concatenates two linked lists

    >>> ll_plus(make_ll(1), make_ll(2,3))
    (1, (2, (3, None)))
    >>> ll_plus(None, make_ll(4,5))
    (4, (5, None))
    """
```

```
def ll_reverse(ll):
    """
    return a new reversed linked list

    >>> ll_reverse(make_ll(1,2,3))
    (3, (2, (1, None)))
    """
```