1. **Lots of Lists**

   Draw the environment diagram that results from executing the following code.

   ```python
   a = [1, 2, 3, 4, 5]
a.pop(3)
b = a[:]
a[1] = b
b[0] = a[:]
b.pop()
b.remove(2)
c = [].append(b[1])
a.insert(b.pop(1), a[-3:4:3])
b.extend(b)
if b == b[:] and b[1][1][0] is b[0][1][1]:
   a, b, c = [c] + a[-4:4:2]
   ```
1. **Tree Recursion with Trees**
   Fill in the function below so that it conforms to its docstring.

   ```python
def about_equal(t1, t2):
    """Returns whether two trees are 'about equal.'
    Two trees are about equal if and only if they contain
    the same labels the same number of times.
    
    >>> x = tree(1, [tree(2), tree(2), tree(3)])
    >>> y = tree(3, [tree(2), tree(1), tree(2)])
    >>> about_equal(x, y)
    True
    >>> z = tree(3, [tree(2), tree(1), tree(2), tree(3)])
    >>> about_equal(x, z)
    False
    """

def label_counts(t):
    if ________________:
        return ________________
    else:
        counts = ________________
        for b in branches(t) + ________________:
            for label, count in ________________:
                if ________________:
                    counts[______________] = 0
                    counts[______________] += ________________
        return counts

    return ________________
```
3 Dictionaries

1. What color is it? (Sp15 Midterm 2 Q4b) Implement `decrypt`, which takes in a string `s` and a dictionary `d` that contains words as values and their secret codes as keys. It returns a list of all possible ways in which `s` can be decoded by splitting it into secret codes and separating the corresponding words by spaces.

```python
def decrypt(s, d):
    """List all possible decoded strings of s."
    >>> codes = {
        ...     'alan': 'spooky',
        ...     'al': 'drink',
        ...     'antu': 'your',
        ...     'turing': 'ghosts',
        ...     'tur': 'scary',
        ...     'ing': 'skeletons',
        ...     'ring': 'ovaltine'
        ...
    }
    >>> decrypt('alanturing', codes)
    ['drink your ovaltine', 'spooky ghosts', 'spooky scary skeletons']
    """

    if s == '':
        return []

    ms = []

    if _________________________________:
        ms.append(__________________________)

    for k in ________________________________:
        first, suffix = s[:k], s[k:]

        if _________________________________:
            for rest in __________________________:
                ms.append(__________________________)

    return ms
```

CS61A Exam Prep Spring 2018: Shea Conlon, Nipun Ramakrishnan and Tiffany Perumpail
2. **Consistency is Key** Fill in the function below so that it conforms to its docstring.

```python
def ensure_consistency(fn):
    """Returns a function that calls fn on its argument, returns fn's return value, and returns None if fn's return value is different from any of its previous return values for those same argument. Also returns None if more than 20 calls are made."
    n = __________
    z = __________
    def helper(x):
        ________
        ________
        if ________:
            return ________
        val = fn(x)
        if ________:
            z[x] = [val]
        if ________:
            return ________
        else:
            z[x] = __________
        return ________
    return helper
```

```python
>>> def consistent(x):
>>>     return x
>>> lst = [1, 2, 3]
>>> def inconsistent(x):
>>>     return x + lst.pop()
>>> a = ensure_consistency(consistent)
>>> a(5) 5
>>> a(5) 5
>>> a(6) 6
>>> a(6) 6
>>> b = ensure_consistency(inconsistent)
>>> b(5) 8
>>> b(5) None
>>> b(6) None
>>> b(6) 7

""
```