Week 4 Discussion Worksheet

1. Given a list of strings ***lst*** and a string ***comparer***, return the number of elements in the given list that are equal to ***comparer***. Your solution must be 1 line. Write assert statements to check the validity of the input.

```
def test(lst, comparer):
    '''
    Throws:
        AssertionError: if lst is not a list
        AssertionError: if comparer is not a string
        AssertionError: if there are non-strings in comparer

    >>> test(['good', 'luck', 'on', 'mideterm', 'luck', 'luck'], 'luck')
    3
    >>> test([], 'lol')
    0
    '''
```

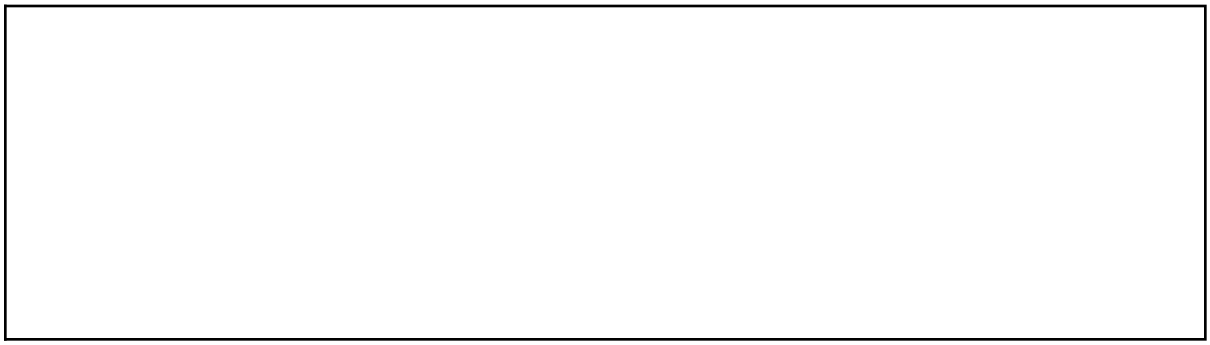
2. Given a dictionary with integers as keys and lists as values, return a new dictionary where the key and value are flipped. The new key would be the length of the value. If the key already exists, add the value to the pre-existing value.

```
def change_dct(input_dct):
    """
    >>> dct = {1:[1,5], 2:[2,6,1], 4:[3,1,5]}
    >>> change_dct(dct)
    {2: 1, 3: 6}
    """
```

3. Given a nested list that contains positive and negative integers, return a nested list that changes negative integers to positive and multiplies positive numbers by 2. You are only allowed to use list comprehension. Write assert statements to check the validity of the input.

```
def convert_negs(lsts):
"""
Throws:
    AssertionError: if lsts is not a list
    AssertionError: if sublists are not lists
    AssertionError: if there are non-integers in sublists

>>> lsts = [[1,3,-11,6], [2,-5,-9,12], [3,19,-42]]
>>> convert_negs(lsts)
[[2, 6, 11, 12], [4, 5, 9, 24], [6, 38, 42]]
"""
```



4. Given a file containing text that's been "graffiti'd", write a function that rewrites the text in the original form (Removes . You may not use .remove() (i.e. the removal must be using string slicing).

```
def de_grafitti(filepath, grafitti):
```