Week 3 Discussion Worksheet

1. What would the values for x, y, z, if I want to print out "summer!" **Note**: Don't forget order of evaluation! (NAO)

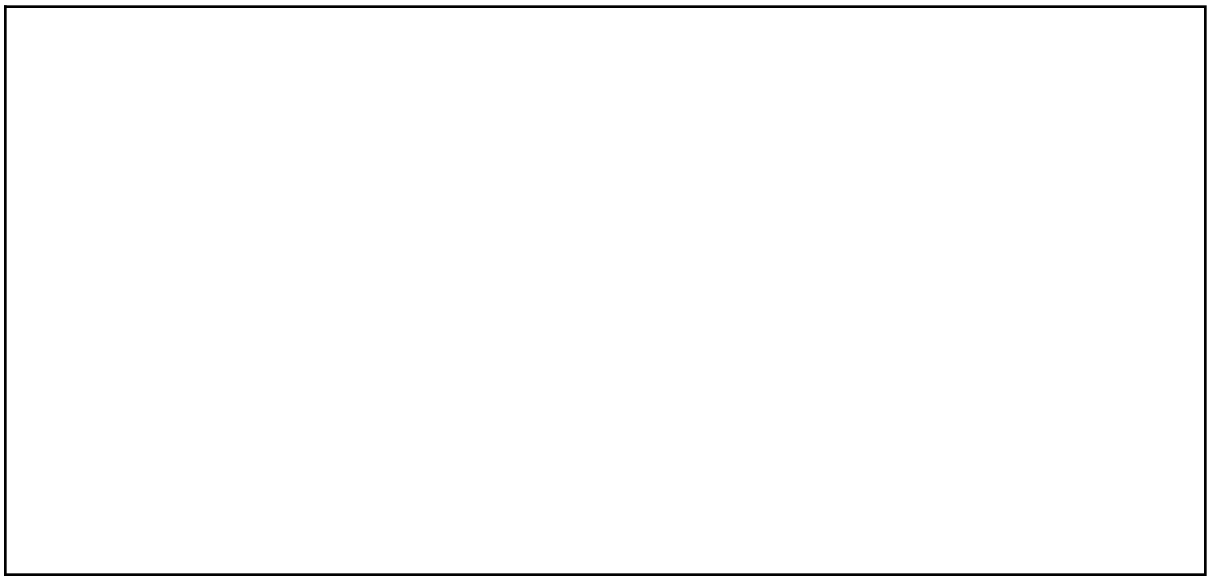
```
if x and y and z:
    print('fall')
elif not x and not y:
    print('winter...')
elif not z or x:
    print('spring?')
else:
    print('summer!')
```

2. Given 2 lists, remove the elements from the first list if they're present in the second one. **Your solution must be 1 line and use filter.**

```
def up_join(lst1, lst2):
    """
    Function that removes elements from lst1 if they're present in lst2.
    Solution must be 1 line and utilize filter

    Args:
        lst1 (list): list of values to be considered
        lst2 (list): list of values to be considered
    Returns:
        a filtered version of lst1

    >>> grades = ['A', 'B', 'C', 'D', 'F']
    >>> grade_filter = ['D', 'F']
    >>> up_join(grades, grade_filter)
    ['A', 'B', 'C']
    """
```

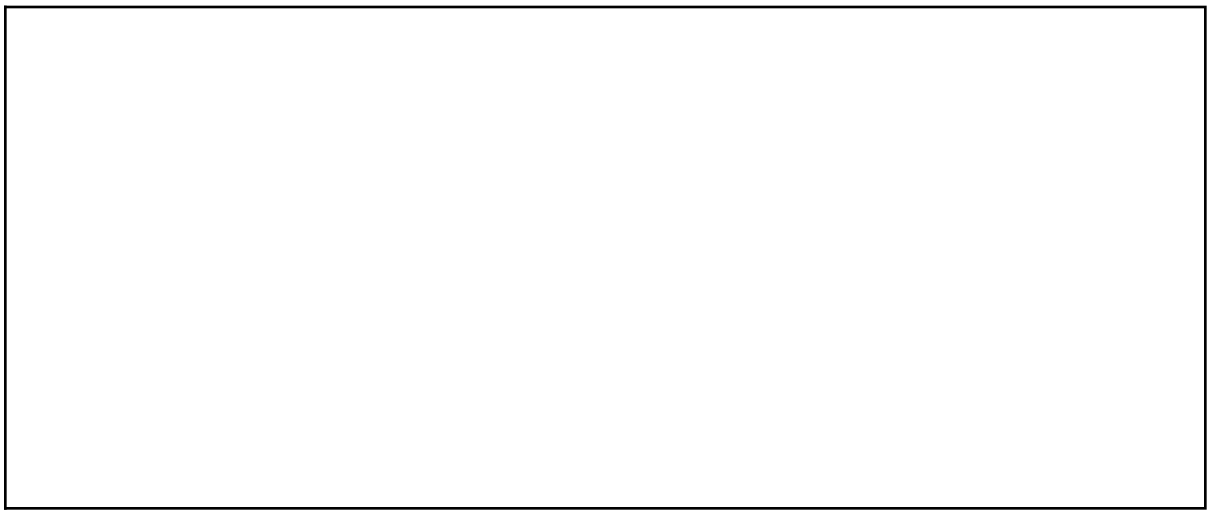
3. Given a dictionary and a string, write a function that creates a new dictionary where the key is the new string and the value is the list of keys who had the new string as their value in the given dictionary. Write assert statements.

```
def flip_dict(owners, pet):
    """
    Throws:
    AssertionError if pet is not a string
    AssertionError if owners is not a dictionary
    AssertionError if the keys and values of owners are not all strings

    >>> sample = {'ben':'cats', 'charisse':'dogs', 'nikki':'cats'}
    >>> flip_dict(sample, 'cats')
    {'cats': ['ben', 'nikki']}
    """
```



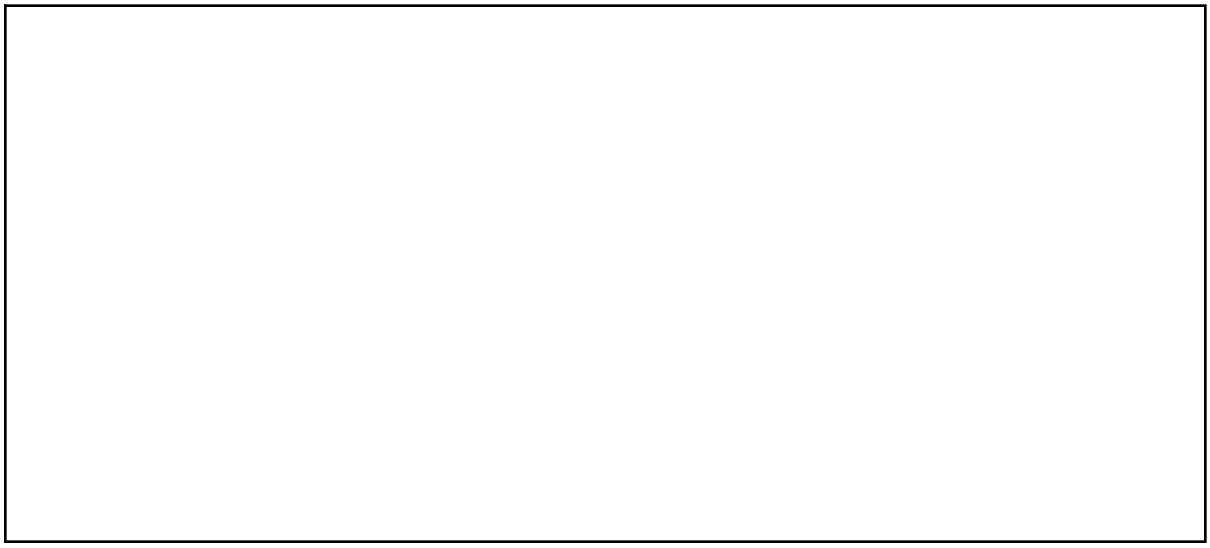
4. Given a file containing an expression on each line, write a function that returns a list which classifies each line as ‘energetic’ if it ends with !, ‘confused’ if it ends with a ?, and ‘neutral’ otherwise.

```
def text_classifier(filepath):
```



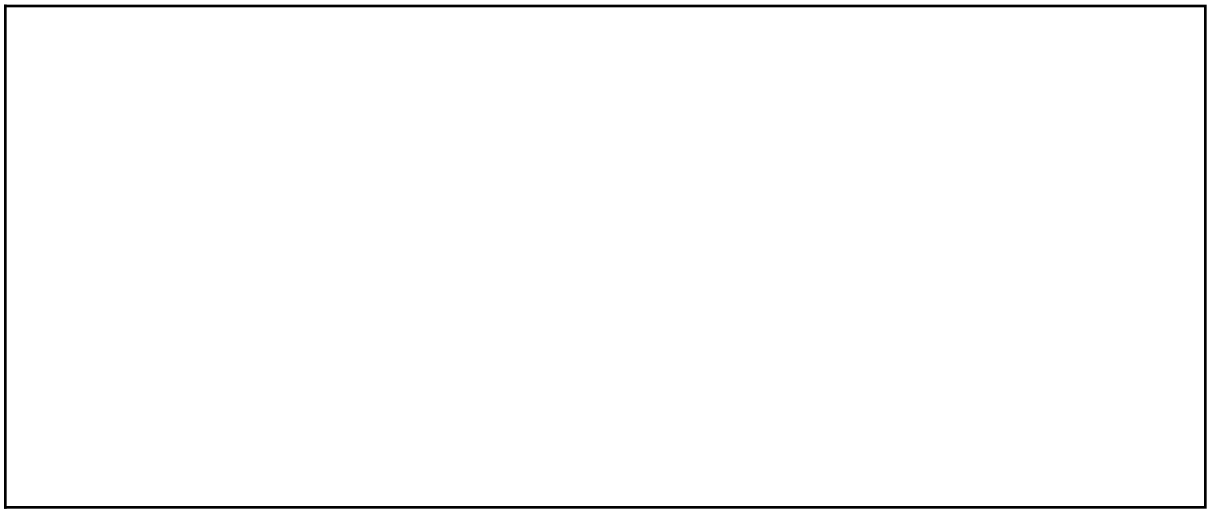
5. Write a function that takes in a list of strings with pattern of first_name, last_name, first_name,... and strings together first and last names with a single space in between. If the length of the list is odd, insert your own last name.

```
def combine_names(names):
    """
    >>> combine_names(['Charisse', 'Hao', 'Nicole', 'Zhang'])
    ['Charisse Hao', 'Nicole Zhang']
    >>> combine_names(['Charisse', 'Hao', 'Ben'])
    ['Charisse Hao','Ben yourlastnamehere']
    """
```



6. Given a dictionary that has lists as values. Write a function that returns a list that consists of the length of each list in the dictionary.

```
def count_values(entries):
    """
    >>> count_values({1: [1,2,3], 2:[3,4,5,6]})
    [3,4]
    """
```



7. Write a function that takes two lists of the same length that contains integers and returns true if the first list is strictly greater than the second list. **One line solution.**

```
def greater_comparison(lst1, lst2):
    """
    >>> greater_comparison([10,20,30], [1,2,3])
    True
    >>> greater_comparison([0,0,4], [1,2,3])
    False
    """
```

8. Write a function that takes in a matrix and a number. It returns the result of the multiplication. **You may only use list comprehension and the solution must be 1 line.**

```
def matrix_multiplication(matrix, coefficient):
    """
    >>> mtx = [[1,2,3],[4,5,6],[7,8,9]]
    >>> matrix_multiplication(mtx, 3)
    [[3, 6, 9], [12, 15, 18], [21, 24, 27]]
    """
```