File: Design.md
Design for all the scrabble_letters methods
Developer: Swathi Danturi

Design for the constructor of scrabble_letters.py

- `self` is the current instance of the class
- `scrabble_file` is the parameter to the constructor passed when the class instance is created
- `self.scrabble_letter` is the instance variable to store the letter frequency and points
- `letters_file` is a file variable used to open the file `scrabble_file` in read mode
- for each line `line` in `letters_file`, iterate through the following:
    - `strip` the `line` to remove any extra spaces or new line
    - `split` the `line` using `,` as delimiter and assign it to the variables `letter`, `count` and `points`
    - if `count` and `points` are digits, do the following:
        - type `count` and `points` to integers
        - append them to a list variable `count_and_points`
    - map the value `count_and_points` list for the key `letter` in the dictionary `self.scrabble_letter`
    - if `letter` is `blank` then the key should be space `' '`
- `self.scrabble_letter` contains letters as the keys and `count` and `points` as value in a list

Design for get_freq method of scrabble_letters.py

- `self` is the current instance of the class
- `letters` string whose frequency is needed is passed as an argument
- intialize `letters_frequency` to an empty dictionary
- for each character `char` in `letters`, iterate through the following:
    - assign the first element in the list values of the dictionary `self.scrabble_letters` at the key `char` to the dictionary `letters_frequency`, at the key `char`
- `letters_frequency` is the dictionary with `letters` as keys and its `frequency` as values
- return `letters_frequency`

Design for reduce_frequency method of scrabble_letters.py

- `self` is the current instance of the class
- `letter` is a character whose frequency is to be reduced is passed as an argument
- convert the upper case `letter` to lower case using `.lower()`
- if the first element in the list of values of the dictionary `self.scrabble_letters` at the key `letter` is greater than zero:
    - reduce it by 1
    - return `True`
- return `False`

Design for get_points method of scrabble_letters.py

- `self` is the current instance of the class
- `word` is a sequence of letters whose sum of points is needed is passed as an argument

- initialize the accumulator `total_points` to zero
- for each character `char` in word, do the following steps:
    - get the `points` of `char` from `self.scrabble_letters` dictionary using `char` as the key
    - use indexing as points is the second element in the `list value` of the key `char`
    - add these `points` to the `total_points`
- return `total_points`