Word Scores

(10 points possible)
The first step is to implement some code that allows us to calculate the score for a single word. The function `getWordScore` should accept as input a string of lowercase letters (a *word*) and return the integer score for that word, using the game's scoring rules.

A Reminder of the Scoring Rules

HINTS

- You may assume that the input `word` is always either a string of lowercase letters, or the empty string `""`.
- You will want to use the `SCRABBLE_LETTER_VALUES` dictionary defined at the top of `ps4a.py`. You should not change its value.
- Do **not** assume that there are always 7 letters in a hand! The parameter `n` is the number of letters required for a bonus score (the maximum number of letters in the hand). Our goal is to keep the code modular - if you want to try playing your word game with *n=10* or *n=4*, you will be able to do it by simply changing the value of `HAND_SIZE`!
- **Testing:** If this function is implemented properly, and you run `test_ps4a.py`, you should see that the `test_getWordScore()` tests pass. Also test your implementation of `getWordScore`, using some reasonable English words.

Fill in the code for `getWordScore` in `ps4a.py` and be sure you've passed the appropriate tests in `test_ps4a.py` before pasting your function definition here.

Canopy specific instructions: If you modify code in `ps4a.py` go to

Run -> Restart Kernel (or hit the CTRL with the dot on your keyboard)

before running `test_ps4a.py` . **You have to do this every time you modify the file** `ps4a.py` **and want to run the file** `test_ps4a.py` , otherwise changes to the former will not be incorporated in the latter.

```
def getWordScore(word, n):
    """
    Returns the score for a word. Assumes the word is a valid word.

    The score for a word is the sum of the points for letters in the
    word, multiplied by the length of the word, PLUS 50 points if all n
    letters are used on the first turn.

    Letters are scored as in Scrabble; A is worth 1, B is worth 3, C is
    worth 3, D is worth 2, E is worth 1, and so on (see SCRABBLE_LETTER_VALUES)

    word: string (lowercase letters)
    n: integer (HAND_SIZE; i.e., hand size required for additional points)
    returns: int >= 0
    """
    # TO DO ... <-- Remove this comment when you code this function
```

Unanswered

You have used 0 of 30 submissions