Problem 1

1 point possible (ungraded)
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

Scoring

- The score for the hand is the sum of the scores for each word formed.

- 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 word created.

- 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. We have defined the dictionary `SCRABBLE_LETTER_VALUES` that maps each lowercase letter to its Scrabble letter value.

- For example, 'weed' would be worth 32 points ((4+1+1+2) for the four letters, then multiply by `len('weed')` to get (4+1+1+2)*4 = 32). Be sure to check that the hand actually has 1 'w', 2 'e's, and 1 'd' before scoring the word!

- As another example, if *n=7* and you make the word 'waybill' on the first try, it would be worth 155 points (the base score for 'waybill' is (4+1+4+3+1+1+1)*7=105, plus an additional 50 point bonus for using all *n* letters).

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.

```
 1 def getWordScore(word, n):
 2     """
 3     Returns the score for a word. Assumes the word is a valid word.
 4
 5     The score for a word is the sum of the points for letters in the
 6     word, multiplied by the length of the word, PLUS 50 points if all n
 7     letters are used on the first turn.
 8
 9     Letters are scored as in Scrabble; A is worth 1, B is worth 3, C is
10     worth 3, D is worth 2, E is worth 1, and so on (see SCRABBLE_LETTER_VALUES)
11
12     word: string (lowercase letters)
13     n: integer (HAND_SIZE; i.e., hand size required for additional points)
14     returns: int >= 0
15     """
16     # TO DO        <-- Remove this comment when you code this function
```

Press ESC then TAB or click outside of the code editor to exit

Unanswered

Submit   You have used 0 of 30 attempts

PSET4: Problem 1

Hide Discussion

Topic: Sandbox / PSET4: Problem 1

Add a Post

Show all posts ⌄   by recent activity ⌄

There are no posts in this topic yet.

✖