You can write code in this Jupyter Notebook to solve the following problems but the solution needs to be in two .py files as described below. Please upload these two files (your **scrabble.py** and **wordscore.py** files) with your solutions to your GitHub repository and gradescope.

Objectives:

- Understand PEP 8 standards
- Use all of your previously gained knowledge together on a single program
- Demonstrate how to import a user-made module and function into python from another .py file
- Demonstrate how to refine an algorithm

6. Cheating at Scrabble (100 points)

Write a Python program that takes a Scrabble rack as a function argument and prints all "valid Scrabble English" words that can be constructed from that rack, along with their Scrabble scores, sorted by score. "valid Scrabble English" words are provided in the data source below. A Scrabble rack is made up of 2 to 7 characters.

Below are the requirements for the program:

- This needs to be able to be run as a function as shown below (not an input statement!)
- Name the python file: `scrabble.py` with the main function inside `scrabble.py` named `run_scrabble`
- Make a separate module named `wordscore.py` which contains, at a minimum, a function called `score_word` . This `score_word` function will take each word and return the score (scoring dictionary is described below). Import this function into your main `scrabble.py` program.
- **NOTE** Only have the function defs and import statements in your `.py` files as the autograder will fail if there are other statements outside the functions.
- Allow anywhere from 2-7 character tiles (letters A-Z, upper or lower case) to be inputted.
- Do not restrict the number of same tiles (e.g., a user is allowed to input ZZZZZQQ).
- Return two items from the `run_scrabble` function:
  - 1) The **total** list of valid Scrabble words that can be constructed from the rack as (score, word) tuples, sorted by the score and then by the word alphabetically as shown in the first example below. All outputted words need to be in upper case.
  - 2) The Total number of valid words as an integer
  - See examples below for the required output. The autograder is looking for this output so please make sure your solution is in the same format shown.
- You need to handle input errors from the user and suggest what that error might be caused by and how to fix it (i.e., a helpful error message). **Return** this error message as a string from the run_scrabble function (do not raise an exception).
- Implement wildcards as either `*` or `?` . There can be a total of **only** two wild cards in any user input (that is, one of each character: one `*` and one `?`). Only use the `*` and `?` as wildcard characters. A wildcard character can take any value A-Z. Replace the wildcard symbol with the letter in your answer (see the second example below).
  - Wildcard characters are scored as 0 points, just like in the real Scrabble game. A word that just consists of two wildcards can be made, should be outputted and scored as 0 points.
  - In a wildcard case where the same word can be made with or without the wildcard, display the highest score. For example: given the input 'I?F', the word 'if' can be made with the wildcard '?F' as well as the letters 'IF'. Since using the letters 'IF' scores higher, display that score.

- For partial credit, your program should take less than one minute to run with 2 wildcards in the input. For full credit, the program needs to run with 2 wildcards in less than 30 seconds.
- Write docstrings for the functions and puts comments in your code.
- You may only use the Python standard library in this assignment. However, any function in the standard library is allowed.

An example invocation and output:

```
run_scrabble("ZAEfiee") -> (
[(17, 'FEAZE'),
(17, 'FEEZE'),
(16, 'FAZE'),
(15, 'FEZ'),
(15, 'FIZ'),
(12, 'ZEA'),
(12, 'ZEE'),
(11, 'ZA'),
(6, 'FAE'),
(6, 'FEE'),
(6, 'FIE'),
(5, 'EF'),
(5, 'FA'),
(5, 'FE'),
(5, 'IF'),
(2, 'AE'),
(2, 'AI'),
(2, 'EA'),
(2, 'EE')],
19
)
```

An example wildcard invocation and output:

```
run_scrabble("?F") -> (
[(4, 'EF'),
(4, 'FA'),
(4, 'FE'),
(4, 'FY'),
(4, 'IF'),
(4, 'OF')],
6
)
```

The Data

The file: http://courses.cms.caltech.edu/cs11/material/advjava/lab1/sowpods.zip or https://drive.google.com/file/d/1ewUiZL_4HanCDsaYB5pcKEgqjMFVgGnh/view?usp=sharing contains all "valid Scrabble English" words in the official words list, one word per line. You should download the word file and keep it in your repository so that the program is standalone (instead of accessing it over the web from Python).

You can read data from a text file with the following code (you can expect sowpods.txt to be run in the same folder as your scrabble.py file):

```
with open("sowpods.txt","r") as infile:
    raw_input = infile.readlines()
    data = [datum.strip('\n') for datum in raw_input]
```

This will show the first 6 words:

```
print(data[0:6])
```

Please use the dictionary below containing the letters and their Scrabble values:

```
scores = {"a": 1, "c": 3, "b": 3, "e": 1, "d": 2, "g": 2,
         "f": 4, "i": 1, "h": 4, "k": 5, "j": 8, "m": 3,
         "l": 1, "o": 1, "n": 1, "q": 10, "p": 3, "s": 1,
         "r": 1, "u": 1, "t": 1, "w": 4, "v": 4, "y": 4,
         "x": 8, "z": 10}
```

Grading Breakdown

You will get 80 points from the autograder for this assignment and 20 points will be hidden. That is, passing all of the visible tests will give you 80 points. Make sure you are meeting the requirements of the problem to get the other 20 points!

- Uploading the correctly named files and functions (5 points)
- Docstrings and comments (5 points)
- User error checking (10 points)
- Code works with no wildcards (30)
- Code works with one wildcard (20)
- Code works with two wildcards (20)
- Algorithm efficiency with 2 wildcards (10 points)

Tips:

- If you don't know what "scrabble" is or the basic background of the game please look it up online!
- We recommend that you try to break down the problem into steps on your own before writing any code. Once you've scoped generally what you want to do, then start writing some code. If you get stuck, go back to thinking about the problem rather than trying to fix lots of errors at the code level.
- If you keep getting stuck, then check out: https://wiki.openhatch.org/wiki/Scrabble_challenge or https://drive.google.com/file/d/1g3yz5ljkzaAeQ-AgQR1Hofy8ZJ0jo25x/view?usp=sharing. This is where we got the idea for this assignment and it provides some helpful tips for guiding you along the way. However, we would recommend that you try to implement this first before looking at the hints on the website.

Good luck!

```
In [ ]:
```