Vowel Position

Write a Python program called `vpos.py` that will accepts exactly two positional arguments:

1. A `vowel`
2. Some `text`

If run with no arguments, the program should print a brief usage:

```
$ ./vpos.py
usage: vpos.py [-h] vowel text
vpos.py: error: the following arguments are required: vowel, text
```

If run with `-h` or `--help`, it should print a longer usage:

```
$ ./vpos.py -h
usage: vpos.py [-h] vowel text

Find position of vowel in string

positional arguments:
  vowel       A vowel to look for
  text        The text to search

optional arguments:
  -h, --help  show this help message and exit
```

The program should reject any `vowel` argument that is not "a," "e," "i," "o," or "u," irrespective of case:

```
$ ./vpos.py b
usage: vpos.py [-h] vowel text
vpos.py: error: argument vowel: invalid choice: 'b' (choose from 'a', 'e', 'i', 'o',
'u', 'A', 'E', 'I', 'O', 'U')
```

The program will search for the `vowel` in the `text`. If it is present, the program will report the *index* in the `text` where the `vowel` can be found:

```
$ ./vpos.py a apple
Found "a" in "apple" at index 0.
$ ./vpos.py E APPLE
Found "E" in "APPLE" at index 4.
```

If the `vowel` is not found in the `text`, report as such:

```
$ ./vpos.py o apple
"o" is not found in "apple".
```

Tests

If you have `make`, you can run `make test`, or you can directly run `pytest -xv test.py` A passing test suite looks like this:

```
$ make test
pytest -xv test.py
============================= test session starts ==============================
...
collected 8 items

test.py::test_exists PASSED                                              [ 12%]
test.py::test_usage PASSED                                               [ 25%]
test.py::test_bad_vowel PASSED                                           [ 37%]
test.py::test_found_a PASSED                                             [ 50%]
test.py::test_found_e PASSED                                             [ 62%]
test.py::test_found_O PASSED                                             [ 75%]
test.py::test_not_found_lower PASSED                                     [ 87%]
test.py::test_not_found_upper PASSED                                     [100%]

============================== 8 passed in 0.42s ===============================
```

Hints

- Start off with `new.py`. That should help you pass the first two tests.
- Read Ch. 2 section "Two different positional arguments" to think about how you can accept `vowel` and `text`.
- Read Ch. 2 section "Restricting values using choices" for an example of restricting valid argument values for the `vowel`.
- Read `help(str)` in the REPL and Ch. 3 to think about string methods you can use to find if one string is in another and how to find the index of a value.
- Think about how you can write conditional branching for when the `vowel` is or is not present in `text`.
- Run the test suite *after every modification to your program*.
- Lean on tools like `pylint` to help you find errors and `yapf` or `black` to format your code to community standards.