

Programming Challenge

This challenge consists of a series of exercises that you should solve using Python 3. You must **submit a single solution.py** module file that has the function implementations for all the exercises.

Keep in mind that there are several ways to solve these exercises, yet many can be considered bad programming. Be thoughtful with your algorithms.

Exercise 1

You must implement a function called `add_between_zeroes`. It receives an integer array, and prints the result of adding all the numbers encompassed by the outermost zeros in the array.

If no zeros are present, it must return 0.

An example execution would look like:

```
$ python
Python 3.6.0 (default, Dec 24 2016, 08:01:42)
Type "help", "copyright", "credits" or "license" for more information.
>>> # solution.py is located in the directory where we executed python
>>> from .solution import add_between_zeroes
>>> add_between_zeroes([3, 4, 0, -2, 0, 7, 4, 0, 8, 7])
9
>>> # It prints 9 since the numbers encompassed by the
>>> # outermost 0s are [-2, 0, 7, 4].
```

Exercise 2

You must write a function `silly_autocomplete`, that receives a string and returns a word from a dictionary that starts with the given string. Your module must load a dictionary file only once and return any word that starts with the given string. Search must be case insensitive. If the given string is shorter than three characters, it must return None. If no suggestions are available for that string, it must return None.

The module must load the dictionary file from the same directory only once when loaded.

The dictionary is available at

<https://raw.githubusercontent.com/jonbcard/scrabble-bot/master/src/dictionary.txt>

An example execution would look like:

```
$ python
Python 3.6.0 (default, Dec 24 2016, 08:01:42)
```

```
Type "help", "copyright", "credits" or "license" for more information.
>>> # solution.py is located in the directory where we executed python
>>> # dictionary.txt is located in the same directory
>>> from .solution import silly_autocomplete
>>> silly_autocomplete("hi")
>>> # Nothing is returned since is shorter than 3 characters
>>> silly_autocomplete("goodby")
"GOODBYE"
>>> silly_autocomplete("GoOdBy")
"GOODBYE"
>>> silly_autocomplete("nomatchforthisword")
>>>
```