The Minion Game

The naïve solution is construct all possible substrings of the input string and categorize them based on their starting letter. This will scale poorly with the length *n* of the input string. Since the substring can start on any of *n* letters and can be up to *n*/2 characters long, on average, we see that this will end up constructing $\mathcal{O}(n^2)$ strings.

We can get around this by realizing that we don't need to actually construct all those strings. The $n-i$ strings that start with letter *i* all will be scored to the same player, so we can just increment their score by this amount. This reduces the $\mathcal{O}(n^2)$ algorithm to $\mathcal{O}(n)$.

```python
def minion_game(string):
    stuart = 0
    kevin = 0
    for index, character in enumerate(string):
        if character in "AEIOU":
            kevin += len(string) - index
        else:
            stuart += len(string) - index
    if stuart > kevin:
        print("Stuart {}".format(stuart))
    elif stuart < kevin:
        print("Kevin {}".format(kevin))
    else:
        print("Draw")
```