1880. Check if Word Equals Summation of Two Words

Easy 123 6 Add to List Share

The **letter value** of a letter is its position in the alphabet **starting from 0** (i.e. `'a' -> 0`, `'b' -> 1`, `'c' -> 2`, etc.).

The **numerical value** of some string of lowercase English letters `s` is the **concatenation** of the **letter values** of each letter in `s`, which is then **converted** into an integer.

- For example, if `s = "acb"`, we concatenate each letter's letter value, resulting in `"021"`. After converting it, we get `21`.

You are given three strings `firstWord`, `secondWord`, and `targetWord`, each consisting of lowercase English letters `'a'` through `'j'` **inclusive**.

Return `true` *if the* ***summation*** *of the* ***numerical values*** *of* `firstWord` *and* `secondWord` *equals the* ***numerical value*** *of* `targetWord`, *or* `false` *otherwise.*

Example 1:

Input: firstWord = "acb", secondWord =

```python
class Solution:
    def isSumEqual(self, firstWord: str, secondWord: str, targetWord: str) -> bool:

        numeric_total = lambda s: int(''.join([str(ord(letter) - ord('a')) for letter in s]))

        return (numeric_total(firstWord) + numeric_total(secondWord) ==numeric_total(targetWord))
```

Testcase Run Code Result Debugger

Accepted Runtime: 44 ms

Your input
"acb"
"cba"

Output true Diff

Expected true

1880/1889

Console Use Example Testcases Run Code Submit