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 = "cba",
targetWord = "cdb"
Output: true
```

```python
class Solution:
    def isSumEqual(self, firstWord: str, secondWord:
str, targetWord: str) -> bool:
        list=
['a','b','c','d','e','f','g','h','i','j']
        for v in firstWord:
            str1= []
            str1.append(list.index(v))
            str11= [str(i) for i in str1]
            f1_str= "".join(str11)

        for v in secondWord:
            str2= []
            str2.append(list.index(v))
            str22= [str(i) for i in str2]
            f2_str= "".join(str22)

        for v in targetWord:
            str3= []
            str3.append(list.index(v))
            str33= [str(i) for i in str3]
            f3_str= "".join(str33)

        if int(f1_str) + int(f2_str) == int(f3_str):
            return True
        return False
```