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

**Explanation:**

The numerical value of firstWord is "acb" -> "021" -> 21.

The numerical value of secondWord is "cba" -> "210" -> 210.

The numerical value of targetWord is "cdb" -> "231" -> 231.

We return true because 21 + 210 == 231.

**Example 2:**

**Input:** firstWord = "aaa", secondWord = "a", targetWord = "aab"

**Output:** false

**Explanation:**

The numerical value of firstWord is "aaa" -> "000" -> 0.

The numerical value of secondWord is "a" -> "0" -> 0.

The numerical value of targetWord is "aab" -> "001" -> 1.

We return false because 0 + 0 != 1.

**Example 3:**

**Input:** firstWord = "aaa", secondWord = "a", targetWord = "aaaa"

**Output:** true

**Explanation:**

The numerical value of firstWord is "aaa" -> "000" -> 0.

The numerical value of secondWord is "a" -> "0" -> 0.

The numerical value of targetWord is "aaaa" -> "0000" -> 0.

We return true because 0 + 0 == 0.

**Constraints:**

* 1 <= firstWord.length, secondWord.length, targetWord.length <= 8
* firstWord, secondWord, and targetWord consist of lowercase English letters from 'a' to 'j' **inclusive**.