



[^ Easy](#)
[🔒 Codewriting](#)
[🏆 1000](#)



A *cryptarithm* is a mathematical puzzle for which the goal is to find the correspondence between letters and digits, such that the given arithmetic equation consisting of letters holds true when the letters are converted to digits.



You have an array of strings `crypt`, the *cryptarithm*, and an array containing the mapping of letters and digits, `solution`. The array `crypt` will contain three non-empty strings that follow the structure: `[word1, word2, word3]`, which should be interpreted as the `word1 + word2 = word3` cryptarithm.



If `crypt`, when it is decoded by replacing all of the letters in the cryptarithm with digits using the mapping in `solution`, becomes a valid arithmetic equation containing no numbers with leading zeroes, the answer is `true`. If it does not become a valid arithmetic solution, the answer is `false`.



Note that number `0` doesn't contain leading zeroes (while for example `00` or `0123` do).

Example

For `crypt = ["SEND", "MORE", "MONEY"]` and

```
solution = [['0', '0'],
            ['M', '1'],
            ['Y', '2'],
            ['E', '5'],
            ['N', '6'],
            ['D', '7'],
            ['R', '8'],
            ['S', '9']]
```

the output should be

```
isCryptSolution(crypt, solution) = true.
```

When you decrypt `"SEND"`, `"MORE"`, and `"MONEY"` using the mapping given in `crypt`, you get `9567 + 1085 = 10652` which is correct and a valid arithmetic equation.

For `crypt = ["TEN", "TWO", "ONE"]` and

```
solution = [['0', '1'],
            ['T', '0'],
            ['W', '9'],
            ['E', '5'],
            ['N', '4']]
```

the output should be

```
isCryptSolution(crypt, solution) = false.
```

Even though `054 + 091 = 145`, `054` and `091` both contain leading zeroes, meaning that this is not a valid solution.

Input/Output

- [execution time limit] 20 seconds (scala)

- [input] array.string crypt

An array of three non-empty strings containing only uppercase English letters.

Guaranteed constraints:

```
crypt.length = 3,
1 ≤ crypt[i].length ≤ 14.
```

- [input] array.array.char solution

An array consisting of pairs of characters that represent the correspondence between letters and numbers in the cryptarithm. The first character in the pair is an uppercase English letter, and the second one is a digit in the range from `0` to `9`.

It is guaranteed that `solution` only contains entries for the letters present in `crypt` and that different letters have different values.

Guaranteed constraints:

```
solution[i].length = 2,
'A' ≤ solution[i][0] ≤ 'Z',
'0' ≤ solution[i][1] ≤ '9',
solution[i][0] ≠ solution[j][0], i ≠ j,
solution[i][1] ≠ solution[j][1], i ≠ j.
```

- [output] boolean

Return `true` if the `solution` represents the correct solution to the cryptarithm `crypt`. otherwise return `false`.

[Scala] Syntax Tips

```
def helloWorld(name: String): String = {  
  println("This prints to the console when you Run Tests")  
  "Hello, " + name  
}
```

⋮ 👍 134 💬 130

300/300

 NEED A CLUE?

SUBMIT

GO TO TOPIC >