Cryptarithm Problem

A cryptarithm is a mathematical puzzle where the goal is to find the correspondence between letters and digits such that the given arithmetic equation consisting of letters holds true.

Given a cryptarithm as an array of strings crypt, count the number of its valid solutions.

Rules for a Valid Solution:

- Each letter must represent a different digit.

- The leading digit of any multi-digit number must not be zero.

The crypt has the following structure:

[word1, word2, word3]

which stands for:

word1 + word2 = word3

Examples

Example 1:

crypt = ["SEND", "MORE", "MONEY"]

Output: solution(crypt) = 1

Explanation:

Only one valid mapping exists:

O = 0, M = 1, Y = 2, E = 5, N = 6, D = 7, R = 8, S = 9

9567 + 1085 = 10652

Example 2:

crypt = ["GREEN", "BLUE", "BLACK"]

Output: solution(crypt) = 12

Explanation: There are 12 valid solutions, such as:

54889 + 6138 = 61027

18559 + 2075 = 20634

72449 + 8064 = 80513

48229 + 5372 = 53601

47119 + 5261 = 52380

36887 + 4028 = 40915

83447 + 9204 = 92651

74665 + 8236 = 82901

65884 + 7308 = 73192

57883 + 6408 = 64291

57881 + 6428 = 64309

83441 + 9264 = 92705

Example 3:

crypt = ["ONE", "TWO", "THREE"]

Output: solution(crypt) = 0

Explanation: No valid solutions.

Input / Output

[execution time limit] 3 seconds (Java)

[memory limit] 1 GB

Input:

- array.string crypt

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

Constraints:

1 <= crypt[i].length <= 35

Output:

- integer

  The number of valid solutions.