```
"""CSP script

  Runs through all possible permutations of the digits 0-9 and assigns them to the l
etters
  in the words FIVE, THREE, and EIGHT. The script then checks if the assignment is a
 valid
  solution to the cryptarithmetic puzzle. If a valid solution is found, the script p
rints
  the solution. If no solution is found, the script prints that no solution exists.
"""

from itertools import permutations

def verify_solution(mapping):
    # convert the words to numbers using the mapping
    FIVE = mapping['F'] * 1000
    + mapping['I'] * 100
    + mapping['V'] * 10
    + mapping['E']

    THREE = mapping['T'] * 10000
    + mapping['H'] * 1000
    + mapping['R'] * 100
    + mapping['E'] * 10
    + mapping['E']

    EIGHT = mapping['E'] * 10000
    + mapping['I'] * 1000
    + mapping['G'] * 100
    + mapping['H'] * 10
    + mapping['T']

    return FIVE + THREE == EIGHT

def solve_csp():
    letters = ['F', 'I', 'V', 'E', 'T', 'H', 'R', 'G']
    digits = range(10) # yields 0-9

    for perm in permutations(digits, len(letters)):
        mapping = dict(zip(letters, perm))

        # F, T, and E must not be 0
        if mapping['F'] == 0 or mapping['T'] == 0 or mapping['E'] == 0:
            continue

        # check if valid
        if verify_solution(mapping):
            return mapping  # return first valid solution

    return None  # no solution


solution = solve_csp()
if solution:
    print("Solution found:")
    for letter, digit in sorted(solution.items()):
        print(f"{letter} = {digit}")
else:
    print("No solution exists.")
```