

# PROGRAM TITLE 3

## CRYPT ARITHMETIC PROBLEM

### AIM:

To write a python program to solve crypt arithmetic problem.

### PROCEDURE:

#### 1. Define Variables:

- Identify the distinct letters in the puzzle and assign them variables. For example, if you have a puzzle like "SEND + MORE = MONEY," assign variables to the letters S, E, N, D, M, O, R, Y.

#### 2. Generate Possible Assignments:

- Use a permutation algorithm to generate all possible assignments of digits to the variables, ensuring that each digit is assigned to a unique letter. You can start with a simple brute-force approach.

#### 3. Evaluate Constraints:

- Implement a function to check whether a given assignment satisfies the constraints of the puzzle. This involves substituting the assigned values into the puzzle equation and verifying that it holds true.

#### 4. Search for Solutions:

- Iterate through the generated assignments and use the constraint evaluation function to identify solutions to the puzzle. Keep track of valid solutions.

#### 5. Print or Output Solutions:

- Once solutions are found, print or output the values of the variables that satisfy the puzzle equation. If there are multiple solutions, you can choose to print all of them.

### CODING:

```
import itertools

def get_value(word, substitution):
    s = 0
    factor = 1
```

```

    for letter in reversed(word):
        s += factor * substitution[letter]
        factor *= 10
    return s

def solve2(equation):

    left, right = equation.lower().replace(' ', '').split('=')

    left = left.split('+')

    letters = set(right)
    for word in left:
        for letter in word:
            letters.add(letter)
    letters = list(letters)
    digits = range(10)
    for perm in itertools.permutations(digits, len(letters)):
        sol = dict(zip(letters, perm))

        if sum(get_value(word, sol) for word in left) == get_value(right, sol):
            print(' + '.join(str(get_value(word, sol)) for word in left) + " = {} (mapping:
            {}).format(get_value(right, sol), sol))

if __name__ == '__main__':
    solve2('SEND + MORE = MONEY')

```

## OUTPUT:

```
7429 + 814 = 8243 (mapping: {'r': 1, 'm': 0, 'y': 3, 'e': 4, 'o': 8, 'n': 2, 'd': 0})
7539 + 815 = 8354 (mapping: {'r': 1, 'm': 0, 'y': 4, 'e': 5, 'o': 8, 'n': 3, 'd': 0})
7649 + 816 = 8465 (mapping: {'r': 1, 'm': 0, 'y': 5, 'e': 6, 'o': 8, 'n': 4, 'd': 0})
8432 + 914 = 9346 (mapping: {'r': 1, 'm': 0, 'y': 6, 'e': 4, 'o': 9, 'n': 3, 'd': 0})
8324 + 913 = 9237 (mapping: {'r': 1, 'm': 0, 'y': 7, 'e': 3, 'o': 9, 'n': 2, 'd': 0})
8542 + 915 = 9457 (mapping: {'r': 1, 'm': 0, 'y': 7, 'e': 5, 'o': 9, 'n': 4, 'd': 0})
6853 + 728 = 7581 (mapping: {'r': 2, 'm': 0, 'y': 1, 'e': 8, 'o': 7, 'n': 5, 'd': 0})
6419 + 724 = 7143 (mapping: {'r': 2, 'm': 0, 'y': 3, 'e': 4, 'o': 7, 'n': 1, 'd': 0})
7531 + 825 = 8356 (mapping: {'r': 2, 'm': 0, 'y': 6, 'e': 5, 'o': 8, 'n': 3, 'd': 0})
7316 + 823 = 8139 (mapping: {'r': 2, 'm': 0, 'y': 9, 'e': 3, 'o': 8, 'n': 1, 'd': 0})
7534 + 825 = 8359 (mapping: {'r': 2, 'm': 0, 'y': 9, 'e': 5, 'o': 8, 'n': 3, 'd': 0})
7643 + 826 = 8469 (mapping: {'r': 2, 'm': 0, 'y': 9, 'e': 6, 'o': 8, 'n': 4, 'd': 0})
5849 + 638 = 6487 (mapping: {'r': 3, 'm': 0, 'y': 7, 'e': 8, 'o': 6, 'n': 4, 'd': 0})
6415 + 734 = 7149 (mapping: {'r': 3, 'm': 0, 'y': 9, 'e': 4, 'o': 7, 'n': 1, 'd': 0})
6524 + 735 = 7259 (mapping: {'r': 3, 'm': 0, 'y': 9, 'e': 5, 'o': 7, 'n': 2, 'd': 0})
6851 + 738 = 7589 (mapping: {'r': 3, 'm': 0, 'y': 9, 'e': 8, 'o': 7, 'n': 5, 'd': 0})
5731 + 647 = 6378 (mapping: {'r': 4, 'm': 0, 'y': 8, 'e': 7, 'o': 6, 'n': 3, 'd': 0})
5732 + 647 = 6379 (mapping: {'r': 4, 'm': 0, 'y': 9, 'e': 7, 'o': 6, 'n': 3, 'd': 0})
3719 + 457 = 4176 (mapping: {'r': 5, 'm': 0, 'y': 6, 'e': 7, 'o': 4, 'n': 1, 'd': 0})
3829 + 458 = 4287 (mapping: {'r': 5, 'm': 0, 'y': 7, 'e': 8, 'o': 4, 'n': 2, 'd': 0})
2817 + 368 = 3185 (mapping: {'r': 6, 'm': 0, 'y': 5, 'e': 8, 'o': 3, 'n': 1, 'd': 0})
2819 + 368 = 3187 (mapping: {'r': 6, 'm': 0, 'y': 7, 'e': 8, 'o': 3, 'n': 1, 'd': 0})
3712 + 467 = 4179 (mapping: {'r': 6, 'm': 0, 'y': 9, 'e': 7, 'o': 4, 'n': 1, 'd': 0})
3821 + 468 = 4289 (mapping: {'r': 6, 'm': 0, 'y': 9, 'e': 8, 'o': 4, 'n': 2, 'd': 0})
9567 + 1085 = 10652 (mapping: {'r': 8, 'm': 1, 'y': 2, 'e': 5, 'o': 0, 'n': 6, 'd': 0})
```

## RESULT:

Hence the program been successfully executed and verified.