PROJECT TITLE -4

Crypt-Arithmetic problem

AIM:

To write and execute the python program for Crypt-Arithmetic problem.

Procedure:

1. Define the Problem:
   - Identify the cryptarithmetic puzzle, including the equation and the letters involved.
2. Generate Possible Assignments:
   - Generate all possible assignments of digits to letters, considering that each letter represents a distinct digit (0-9).
3. Check Constraints:
   - For each assignment, substitute the digits into the equation.
   - Check if the equation is satisfied.
4. Backtracking:
   - If the equation is not satisfied, backtrack and try a different assignment.
   - Continue until a solution is found or all possible assignments are exhausted.
5. Optimizations:
   - Apply heuristics or optimizations to reduce the search space and improve efficiency.

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):

  # split equation in left and right
```

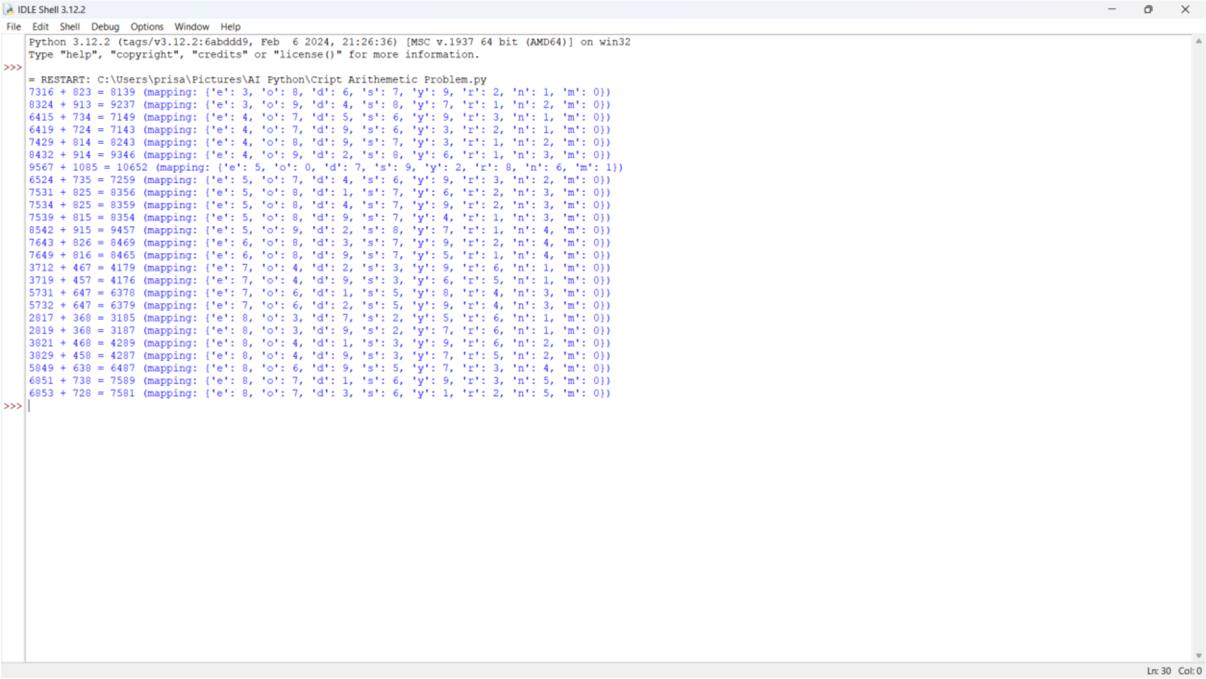
```
    left, right = equation.lower().replace(' ', '').split('=')
    # split words in left part
    left = left.split('+')
    # create list of used letters
    letters = set(right)
    for word in left:
        for letter in word:
            letters.add(letter)
    letters = list(letters)

    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:



```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:\Users\prisa\Pictures\AI Python\Cript Arithemetic Problem.py
7316 + 823 = 8139 (mapping: {'e': 3, 'o': 8, 'd': 6, 's': 7, 'y': 9, 'r': 2, 'n': 1, 'm': 0})
8324 + 913 = 9237 (mapping: {'e': 3, 'o': 9, 'd': 4, 's': 8, 'y': 7, 'r': 1, 'n': 2, 'm': 0})
6415 + 734 = 7149 (mapping: {'e': 4, 'o': 7, 'd': 5, 's': 6, 'y': 9, 'r': 3, 'n': 1, 'm': 0})
6419 + 724 = 7143 (mapping: {'e': 4, 'o': 7, 'd': 9, 's': 6, 'y': 3, 'r': 2, 'n': 1, 'm': 0})
7429 + 814 = 8243 (mapping: {'e': 4, 'o': 8, 'd': 9, 's': 7, 'y': 3, 'r': 1, 'n': 2, 'm': 0})
8432 + 914 = 9346 (mapping: {'e': 4, 'o': 9, 'd': 2, 's': 8, 'y': 6, 'r': 1, 'n': 3, 'm': 0})
9567 + 1085 = 10652 (mapping: {'e': 5, 'o': 0, 'd': 7, 's': 9, 'y': 2, 'r': 8, 'n': 6, 'm': 1})
6524 + 735 = 7259 (mapping: {'e': 5, 'o': 7, 'd': 4, 's': 6, 'y': 9, 'r': 3, 'n': 2, 'm': 0})
7531 + 825 = 8356 (mapping: {'e': 5, 'o': 8, 'd': 1, 's': 7, 'y': 6, 'r': 2, 'n': 3, 'm': 0})
7534 + 825 = 8359 (mapping: {'e': 5, 'o': 8, 'd': 4, 's': 7, 'y': 9, 'r': 2, 'n': 3, 'm': 0})
7539 + 815 = 8354 (mapping: {'e': 5, 'o': 8, 'd': 9, 's': 7, 'y': 4, 'r': 1, 'n': 3, 'm': 0})
8542 + 915 = 9457 (mapping: {'e': 5, 'o': 9, 'd': 2, 's': 8, 'y': 7, 'r': 1, 'n': 4, 'm': 0})
7643 + 826 = 8469 (mapping: {'e': 6, 'o': 8, 'd': 3, 's': 7, 'y': 9, 'r': 2, 'n': 4, 'm': 0})
7649 + 816 = 8465 (mapping: {'e': 6, 'o': 8, 'd': 9, 's': 7, 'y': 5, 'r': 1, 'n': 4, 'm': 0})
3712 + 467 = 4179 (mapping: {'e': 7, 'o': 4, 'd': 2, 's': 3, 'y': 9, 'r': 6, 'n': 1, 'm': 0})
3719 + 457 = 4176 (mapping: {'e': 7, 'o': 4, 'd': 9, 's': 3, 'y': 6, 'r': 5, 'n': 1, 'm': 0})
5731 + 647 = 6378 (mapping: {'e': 7, 'o': 6, 'd': 1, 's': 5, 'y': 8, 'r': 4, 'n': 3, 'm': 0})
5732 + 647 = 6379 (mapping: {'e': 7, 'o': 6, 'd': 2, 's': 5, 'y': 9, 'r': 4, 'n': 3, 'm': 0})
2817 + 368 = 3185 (mapping: {'e': 8, 'o': 3, 'd': 7, 's': 2, 'y': 5, 'r': 6, 'n': 1, 'm': 0})
2819 + 368 = 3187 (mapping: {'e': 8, 'o': 3, 'd': 9, 's': 2, 'y': 7, 'r': 6, 'n': 1, 'm': 0})
3821 + 468 = 4289 (mapping: {'e': 8, 'o': 4, 'd': 1, 's': 3, 'y': 9, 'r': 6, 'n': 2, 'm': 0})
3829 + 458 = 4287 (mapping: {'e': 8, 'o': 4, 'd': 9, 's': 3, 'y': 7, 'r': 5, 'n': 2, 'm': 0})
5849 + 638 = 6487 (mapping: {'e': 8, 'o': 6, 'd': 9, 's': 5, 'y': 7, 'r': 3, 'n': 4, 'm': 0})
6851 + 738 = 7589 (mapping: {'e': 8, 'o': 7, 'd': 1, 's': 6, 'y': 9, 'r': 3, 'n': 5, 'm': 0})
6853 + 728 = 7581 (mapping: {'e': 8, 'o': 7, 'd': 3, 's': 6, 'y': 1, 'r': 2, 'n': 5, 'm': 0})
>>> 
```

Result:

Thus the program has been successfully executed and verified.