**PROJECT TITLE -4**

**Cript-Arithmetic problem**

**AIM**:

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

**Procedure:**

1. **Input Puzzle**: Receive the cryptarithmetic puzzle as input. The puzzle consists of an equation where letters represent digits.
2. **Identify Letters**: Extract all unique letters from the puzzle equation. These letters represent digits that need to be assigned unique values.
3. **Generate Permutations**: Generate all possible permutations of the digits (0-9) for the identified letters. Each permutation represents a possible assignment of digits to letters.
4. **Check Validity**:
   * For each permutation, substitute the digits into the puzzle equation and evaluate it.
   * Evaluate the equation using standard arithmetic rules.
   * If the equation holds true, return the assignment of digits to letters as the solution.
   * If none of the permutations yield a valid solution, return "No solution exists."
5. **Output Solution**: Display the solution, i.e., the assignment of digits to letters that satisfies the puzzle's equation.

**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))

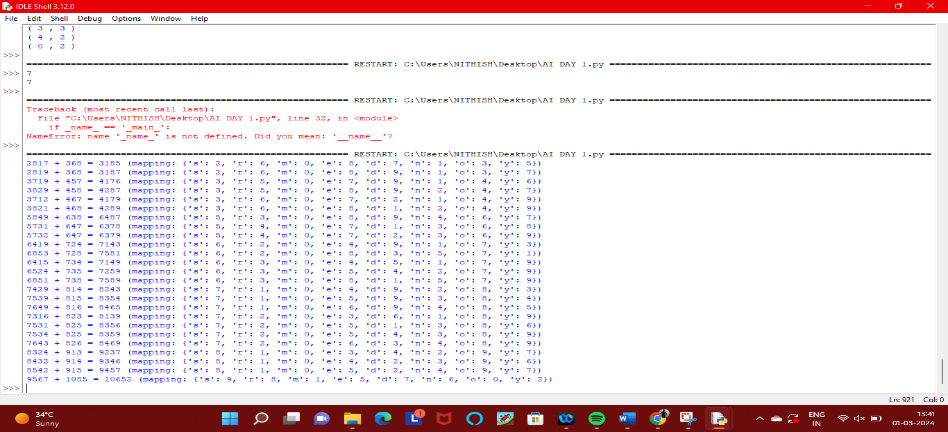
if sum(get\_value(word, sol) for word in left) == get\_value(right, sol):

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:**



**Result:**

Thus the program has been successfully executed and verified.