EXPERIMENT:4

Cript-Arithmetic problem

AIM: To write the python program for Cript-Arithmetic problem

ALGORITHM:

1. Extract unique letters from the puzzle input.
2. Generate permutations of digits from 0 to 9.
3. For each permutation:
   * Skip permutations with leading zeros.
   * Create a mapping from letters to digits using the permutation.
   * Evaluate the puzzle equation with the current mapping.
4. If the puzzle evaluates to true with any permutation, return the mapping.
5. If no solution is found after trying all permutations, return None.

CODE:

from itertools import permutations

def solve\_cryptarithmetic(puzzle):

# Extracting unique letters from the puzzle

letters = set(ch for word in puzzle.split() for ch in word)

# Generating permutations of digits from 0 to 9

for digits in permutations(range(10), len(letters)):

if any(digit == 0 for digit in digits):

continue # Skip permutations with leading zeros

# Creating a mapping from letters to digits

mapping = dict(zip(letters, digits))

# Evaluating the puzzle with the current mapping

if evaluate\_puzzle(puzzle, mapping):

return mapping

return None

def evaluate\_puzzle(puzzle, mapping):

# Replace letters with corresponding digits and evaluate the expression

expression = ''.join(str(mapping[ch]) if ch.isalpha() else ch for ch in puzzle)

left, right = expression.split('=')

left = left.strip()

right = right.strip()

return eval(left) == eval(right)

# Example usage

puzzle = "SEND + MORE = MONEY"

solution = solve\_cryptarithmetic(puzzle)

if solution:

print("Solution found:")

for letter, digit in sorted(solution.items()):

print(f"{letter}: {digit}")

else:

print("No solution found.")

INPUT: "SEND + MORE = MONEY"

OUTPUT: Solution found:

D: 7

E: 5

M: 1

N: 6

O: 0

R: 8

S: 9

Y: 2