from itertools import permutations

def solve\_cryptarithmetic():

# Unique characters in the puzzle

letters = 'SENDMORY'

# Ensure the letters are unique

assert len(set(letters)) == len(letters)

# Try all permutations of 8 digits out of 0-9

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

letter\_map = dict(zip(letters, perm))

# Skip if leading letters are assigned to 0

if letter\_map['S'] == 0 or letter\_map['M'] == 0:

continue

# Construct numbers from words

send = 1000 \* letter\_map['S'] + 100 \* letter\_map['E'] + 10 \* letter\_map['N'] + letter\_map['D']

more = 1000 \* letter\_map['M'] + 100 \* letter\_map['O'] + 10 \* letter\_map['R'] + letter\_map['E']

money = 10000 \* letter\_map['M'] + 1000 \* letter\_map['O'] + 100 \* letter\_map['N'] + 10 \* letter\_map['E'] + letter\_map['Y']

# Check if the equation SEND + MORE == MONEY is satisfied

if send + more == money:

print(f"SEND = {send}, MORE = {more}, MONEY = {money}")

print("Solution:", letter\_map)

return

print("No solution found.")

# Run the solver

solve\_cryptarithmetic()

OUTPUT:

