**Experiment 3:**

**Aim:** To write a python program to implement a Crypt Arithemetic program.

**Algorithm:**

1. Represent the equation as word-based arithmetic with unique letters as digits (0–9, no duplicates).
2. Extract unique letters and generate all digit permutations for them.
3. Ensure no leading zeros and evaluate the equation by substituting digits for letters.
4. Record mappings where the equation holds true as valid solutions.
5. Display all valid solutions or indicate if none exist.

**Program:**

import itertools

def is\_valid\_solution(mapping, left\_expr, right\_expr):

"""

Check if the given mapping satisfies the cryptarithm.

"""

# Replace letters with corresponding digits

for letter, digit in mapping.items():

left\_expr = left\_expr.replace(letter, str(digit))

right\_expr = right\_expr.replace(letter, str(digit))

# Leading zero check

if any(num.startswith('0') for num in left\_expr.split('+')):

return False

if right\_expr.startswith('0'):

return False

# Evaluate the expressions

try:

return eval(left\_expr) == eval(right\_expr)

except:

return False

def solve\_cryptarithm(cryptarithm):

"""

Solve the cryptarithm puzzle.

"""

# Split into left and right expressions

cryptarithm = cryptarithm.replace(" ", "")

if '=' not in cryptarithm:

print("Invalid format! Please include an '=' symbol.")

return

left\_expr, right\_expr = cryptarithm.split('=')

# Extract unique letters

unique\_letters = set(filter(str.isalpha, cryptarithm))

if len(unique\_letters) > 10:

print("Too many unique letters (maximum is 10).")

return

# Try all permutations of digits for the letters

for perm in itertools.permutations(range(10), len(unique\_letters)):

mapping = dict(zip(unique\_letters, perm))

if is\_valid\_solution(mapping, left\_expr, right\_expr):

print("\nSolution found:")

for letter, digit in mapping.items():

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

return

print("No solution found.")

def main():

print("Cryptarithm Solver")

print("Example: SEND + MORE = MONEY")

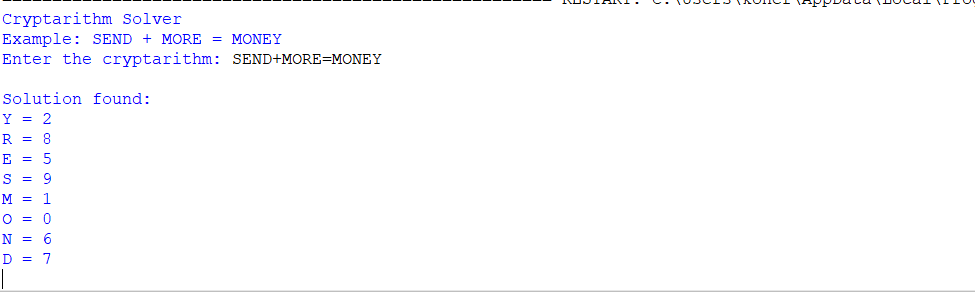
cryptarithm = input("Enter the cryptarithm: ")

solve\_cryptarithm(cryptarithm)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**OUTPUT:**

****

**Result:** Thus, the program was successfully completed using python programming.