```
def find_value(word, assigned):
  num = 0
  for char in word:
    num = num * 10
    num += assigned[char]
  return num


def is_valid_assignment(word1, word2, result, assigned):
  # First letter of any word cannot be zero.
  if assigned[word1[0]] == 0 or assigned[word2[0]] == 0 or assigned[
      result[0]] == 0:
    return False
  return True


def _solve(word1, word2, result, letters, assigned, solutions):
  if not letters:
    if is_valid_assignment(word1, word2, result, assigned):
      num1 = find_value(word1, assigned)
      num2 = find_value(word2, assigned)
      num_result = find_value(result, assigned)
      if num1 + num2 == num_result:
        solutions.append((f'{num1} + {num2} = {num_result}', assigned.copy()))
    return

  for num in range(10):
    if num not in assigned.values():
      cur_letter = letters.pop()
      assigned[cur_letter] = num
      _solve(word1, word2, result, letters, assigned, solutions)
      assigned.pop(cur_letter)
      letters.append(cur_letter)


def solve(word1, word2, result):
  letters = sorted(set(word1) | set(word2) | set(result))
  if len(result) > max(len(word1), len(word2)) + 1 or len(letters) > 10:
    print('0 Solutions!')
    return

  solutions = []
  _solve(word1, word2, result, letters, {}, solutions)
  if solutions:
    print('\nSolutions:')
    for soln in solutions:
      print(f'{soln[0]}\t{soln[1]}')


print('CRYPTARITHMETIC PUZZLE SOLVER')
print('WORD1 + WORD2 = RESULT')
word1 = input('Enter WORD1: ').upper()
word2 = input('Enter WORD2: ').upper()
result = input('Enter RESULT: ').upper()

if not word1.isalpha() or not word2.isalpha() or not result.isalpha():
    raise TypeError('Inputs should ony consists of alphabets.')

solve(word1, word2, result)
```

```
CRYPTARITHMETIC PUZZLE SOLVER
WORD1 + WORD2 = RESULT
Enter WORD1: SEND
Enter WORD2: MORE
Enter RESULT: MONEY

Solutions:
9567 + 1085 = 10652     {'Y': 2, 'S': 9, 'R': 8, 'O': 0, 'N': 6, 'M': 1, 'E': 5, 'D': 7}
```

Start coding or generate with AI.