import itertools

def puzzle(word1, word2,word3):

#Converting words to lowercase to deal with case sensitivitity

l1 = word1.lower() + word2.lower()

r1 = word3.lower()

#Creating set to avoid dupliacte characters

letters = set(r1)

#Creating list for storing left side of the words

mainlist = list()

mainlist.append(word1.lower())

mainlist.append(word2.lower())

for word in l1:

for letter in word:

letters.add(letter)

#AStoring all unique charactes in list

letters = list(letters)

print("Unique Letters:", letters)

digits = range(10) #Since characters can be between 0-9

#Finding all the permutations of letters and thier numeric values and storing it in dictionary

for p in itertools.permutations(digits, len(letters)):

subs = dict(zip(letters, p))

#if the sum of LHS = RHS

if sum(get\_value(word, subs) for word in mainlist) == get\_value(r1, subs):

print(' + '.join(str(get\_value(word, subs)) for word in mainlist) + " = {} \nMAP {}".format(get\_value(r1, subs), subs))

def get\_value(word, substitution):

s = 0

factor = 1

for letter in reversed(word):

#The sum operation for each factors

s += factor \* substitution[letter]

factor \*= 10

return s

puzzle('UBER', 'COOL' , 'UNCLE')