ARTIFICIAL INTELLIGENCE

LAB 3

Sahil Satyam
RA1911003010675

AIM : Cryptarithmetic Problem.

PROBLEM FORMULATION :

Cryptarithmetic problem is a type of constraint satisfaction problem, where the game is about digits and its unique replacement either with alphabets or other symbols. The task of cryptarithmetic problem is to substitute each digit with an alphabet to get the result arithmetically correct.

PROBLEM SOLVING :

- SEND + MORE = MONEY

→ starting from the LHS, the terms are S and M. Assign a digit which could give a satisfactory result.
let's assign S → 9 & M → 1.

```
  S        9
+ M      + 1
----     ----
 M O      1 0        O → 0.
```

→ now move ahead to the next terms E and O to get N as its output.

```
  E                 5
+ O    —X→        + 0
----              ----
  N                 5       <not possible to assign
                             5 to both E & N>
```

(1) we take.

→
$$\begin{array}{r} E \\ +0 \\ \hline N \end{array} \qquad \begin{array}{r} 5 \\ +0 \\ \hline 6 \end{array}$$ ← carry.

E=5 N=6

→ Further, adding N & R.

$$\begin{array}{r} N \\ +R \\ \hline E \end{array} \rightarrow \begin{array}{r} 6 \\ +8 \\ \hline 14 \end{array}$$

but we have assigned E→5. So,

$$\begin{array}{r} N \\ +R \\ \hline E \end{array} \qquad \begin{array}{r} (1)\ 6 \\ +8 \\ \hline 15 \end{array}$$ ← carry.

↑ <one will be carry forward to the above term>.

→
$$\begin{array}{r} D \\ +E \\ \hline Y \end{array} \qquad \begin{array}{r} 7 \\ +5 \\ \hline 12 \end{array}$$

↑
<one will be carry forward to the above term>.

the final result will be.

$$\begin{array}{r} SEND \\ +MORE \\ \hline MONEY \end{array} \Leftrightarrow \begin{array}{r} 9567 \\ +1085 \\ \hline 10652 \end{array}$$

therefore

S: 9	M: 1	M: 1
E: 5	O: 0	O: 0
N: 6	R: 8	N: 6
D: 7	E: 5	E: 5
		Y: 2

ALGORITHM:

- start
- accept an expression "SEND + MORE = MONEY".
- extract the operands from the equation.
- permute for different combination of values for S, E, N, D, M, O, R, Y
- check if the sum of the left value i.e. SEND + MORE is equal to the right sum i.e. MONEY.
- If the sum value matches map the digits for that alphabet.
- continue for other alphabets.
- stop.

Program :

```
import itertools
def get_value(word, substitution):
    s = 0
    factor = 1
    for letter in reversed(word):
        s += factor * substitution[letter]
        factor *= 10
    return s
def solve(equation):
    left, right = equation.lower().replace(' ', '').split('=')
    left = left.split('+')
    letters = set(right)
    for word in left:
        for letter in word:
            letters.add(letter)
    letters = list(letters)

    digits = range(10)
    for perm in itertools.permutations(digits, len(letters)):
        sol = dict(zip(letters, perm))

        if sum(get_value(word, sol) for word in left) == get_value(right, sol):
            print(' + '.join(str(get_value(word, sol)) for word in left) + " = {} (mapping:
{})".format(get_value(right, sol), sol))
solve('BASE+BALL=GAMES')
```

Output: