Akshat Agarwal
RA1911003010646

Artificial Intelligence
Lab - 3

Aim : Implementation of Constraint Satisfaction Problems, i.e., Cryptarithmetic Problem- SEND + MORE = MONEY

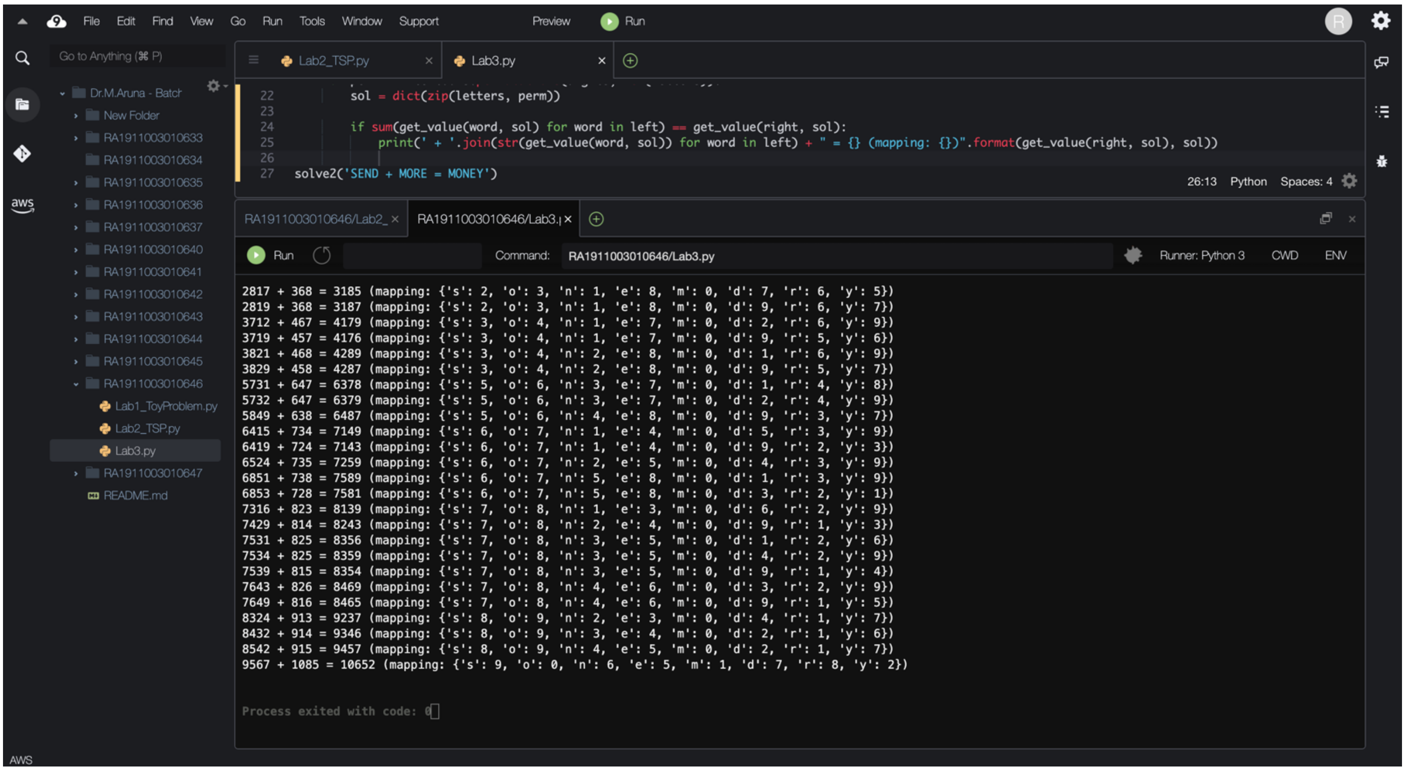
Algorithm :

1. Start
2. Accept an expression ‘SEND+MORE=MONEY’
3. Extract the words SEND,MORE and MONEY.
4. Permute for different combination of values for S,E,N,D,M,O,R,Y.
5. And check if the sum of left value i.e., SEND+MORE is equal to right sum i.e., MONEY or not. If the sum value matches print the mapping.
6. Continue for other permutations as well.
7. Stop.

Code :

```
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 solve2(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))

    solve2('SEND + MORE = MONEY')
```

Output :



```
        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))

solve2('SEND + MORE = MONEY')
```

```
2817 + 368 = 3185 (mapping: {'s': 2, 'o': 3, 'n': 1, 'e': 8, 'm': 0, 'd': 7, 'r': 6, 'y': 5})
2819 + 368 = 3187 (mapping: {'s': 2, 'o': 3, 'n': 1, 'e': 8, 'm': 0, 'd': 9, 'r': 6, 'y': 7})
3712 + 467 = 4179 (mapping: {'s': 3, 'o': 4, 'n': 1, 'e': 7, 'm': 0, 'd': 2, 'r': 6, 'y': 9})
3719 + 457 = 4176 (mapping: {'s': 3, 'o': 4, 'n': 1, 'e': 7, 'm': 0, 'd': 9, 'r': 5, 'y': 6})
3821 + 468 = 4289 (mapping: {'s': 3, 'o': 4, 'n': 2, 'e': 8, 'm': 0, 'd': 1, 'r': 6, 'y': 9})
3829 + 458 = 4287 (mapping: {'s': 3, 'o': 4, 'n': 2, 'e': 8, 'm': 0, 'd': 9, 'r': 5, 'y': 7})
5731 + 647 = 6378 (mapping: {'s': 5, 'o': 6, 'n': 3, 'e': 7, 'm': 0, 'd': 1, 'r': 4, 'y': 8})
5732 + 647 = 6379 (mapping: {'s': 5, 'o': 6, 'n': 3, 'e': 7, 'm': 0, 'd': 2, 'r': 4, 'y': 9})
5849 + 638 = 6487 (mapping: {'s': 5, 'o': 6, 'n': 4, 'e': 8, 'm': 0, 'd': 9, 'r': 3, 'y': 7})
6415 + 734 = 7149 (mapping: {'s': 6, 'o': 7, 'n': 1, 'e': 4, 'm': 0, 'd': 5, 'r': 3, 'y': 9})
6419 + 724 = 7143 (mapping: {'s': 6, 'o': 7, 'n': 1, 'e': 4, 'm': 0, 'd': 9, 'r': 2, 'y': 3})
6524 + 735 = 7259 (mapping: {'s': 6, 'o': 7, 'n': 2, 'e': 5, 'm': 0, 'd': 4, 'r': 3, 'y': 9})
6851 + 738 = 7589 (mapping: {'s': 6, 'o': 7, 'n': 5, 'e': 8, 'm': 0, 'd': 1, 'r': 3, 'y': 9})
6853 + 728 = 7581 (mapping: {'s': 6, 'o': 7, 'n': 5, 'e': 8, 'm': 0, 'd': 3, 'r': 2, 'y': 1})
7316 + 823 = 8139 (mapping: {'s': 7, 'o': 8, 'n': 1, 'e': 3, 'm': 0, 'd': 6, 'r': 2, 'y': 9})
7429 + 814 = 8243 (mapping: {'s': 7, 'o': 8, 'n': 2, 'e': 4, 'm': 0, 'd': 9, 'r': 1, 'y': 3})
7531 + 825 = 8356 (mapping: {'s': 7, 'o': 8, 'n': 3, 'e': 5, 'm': 0, 'd': 1, 'r': 2, 'y': 6})
7534 + 825 = 8359 (mapping: {'s': 7, 'o': 8, 'n': 3, 'e': 5, 'm': 0, 'd': 4, 'r': 2, 'y': 9})
7539 + 815 = 8354 (mapping: {'s': 7, 'o': 8, 'n': 3, 'e': 5, 'm': 0, 'd': 9, 'r': 1, 'y': 4})
7643 + 826 = 8469 (mapping: {'s': 7, 'o': 8, 'n': 4, 'e': 6, 'm': 0, 'd': 3, 'r': 2, 'y': 9})
7649 + 816 = 8465 (mapping: {'s': 7, 'o': 8, 'n': 4, 'e': 6, 'm': 0, 'd': 9, 'r': 1, 'y': 5})
8324 + 913 = 9237 (mapping: {'s': 8, 'o': 9, 'n': 2, 'e': 3, 'm': 0, 'd': 4, 'r': 1, 'y': 7})
8432 + 914 = 9346 (mapping: {'s': 8, 'o': 9, 'n': 3, 'e': 4, 'm': 0, 'd': 2, 'r': 1, 'y': 6})
8542 + 915 = 9457 (mapping: {'s': 8, 'o': 9, 'n': 4, 'e': 5, 'm': 0, 'd': 2, 'r': 1, 'y': 7})
9567 + 1085 = 10652 (mapping: {'s': 9, 'o': 0, 'n': 6, 'e': 5, 'm': 1, 'd': 7, 'r': 8, 'y': 2})


Process exited with code: 0
```

RA1911003010646
Akshat Agarwal

AI Lab-3

Aim: To implement CSP, i.e., Cryptarithmetic problem SEND + MORE = MONEY

Problem Formulation

Given an expression where two words adds to give a third word, assign some unique digit (0-9) to each letter such that the addition of the two words make the third word. Display the possible mappings for the same.

Initial State

- S = ?, E = ?, N = ?, D = ?, M = ?, O = ?, R = ?, Y = ?
- C_1 = ?, C_2 = ?, C_3 = ? (Carry Variables)

Goal State

In the goal state the no. assigned to the letters should satisfy the sum.

eg:

- D = 7, E = 5, Y = 2, N = 6, R = 8, O = 0, S = 9, M = 1
- C_1 = 1, C_2 = 1, C_3 = 0

Problem Solving

- We will start from LHS

$$\begin{array}{r} S \\ +\,M \\ \hline MO \end{array} \qquad \begin{array}{r} 9 \\ +\,1 \\ \hline 10 \end{array}$$

- Moving ahead, Consider E = 5

$$\begin{array}{r} E \\ +\,O \\ \hline N \end{array}$$

As O is zero, there is some carry factor as E & N can't be same

$$\begin{array}{r} C_2 \\ E \\ +\,O \\ \hline N \end{array} \qquad \begin{array}{r} 1 \\ 5 \\ +\,O \\ \hline 6 \end{array}$$

$\therefore N = 6$

- Moving ahead,

$$\begin{array}{r} N \\ +\,R \\ \hline E \end{array} \rightarrow \begin{array}{r} C_3 \\ N \\ +\,R \\ \hline E \end{array}$$

$$\Rightarrow \begin{array}{r} 1 \\ 6 \\ +\,8 \\ \hline 15 \end{array}$$

- On adding last two terms,

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

<u>Final State</u>

$$\begin{array}{cccccc} & 0_{c_3} & 1_{c_2} & 1_{c_1} & \\ & \boxed{9}_S & \boxed{5}_E & \boxed{6}_N & \boxed{7}_D \\ + & \boxed{1}_M & \boxed{0}_O & \boxed{8}_R & \boxed{5}_E \\ \hline \boxed{1}_M & \boxed{0}_O & \boxed{6}_N & \boxed{5}_E & \boxed{2}_Y \end{array}$$