

LAB 3 – IMPLEMENTATION OF CSP

AI LAB

SHUBHAM SHARMA

RA1911003010649

ALGORITHM:

Step 1: Start

Step 2: Accept an expression 'SEND+MORE=MONEY'.

Step 3: Extract the words SEND, MORE and MONEY.

Step 4: Permute for different combinations of values for S, E, N, D, M, O, R, Y.

Step 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.

Step 6: Continue for other permutations as well.

Step 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))
```

```
if __name__ == '__main__':
```

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

OUTPUT:

```
1 import itertools
2 def get_value(word, substitution):
3     s = 0
4     factor = 1
5     for letter in reversed(word):
6         s = factor * substitution[letter]
7         factor *= 10
8
9 RA1911003010658/10.cc x RA1911003010658/py.py x RA1911003010658/TSP.py
10 Run Python Spaces 4
11 Command RA1911003010658/py.py
12 Runner Python 3 CWD ENV
13 3821 + 468 = 4289 (mapping: {'d': 1, 'n': 2, 'e': 6, 's': 8, 't': 3, 'm': 0, 'y': 9, 'o': 4})
14 7511 + 825 = 8336 (mapping: {'d': 1, 'n': 3, 'e': 2, 's': 5, 't': 7, 'm': 0, 'y': 6, 'o': 8})
15 5731 + 647 = 6378 (mapping: {'d': 1, 'n': 3, 'e': 4, 's': 7, 't': 5, 'm': 0, 'y': 8, 'o': 6})
16 6851 + 738 = 7589 (mapping: {'d': 1, 'n': 5, 'e': 3, 's': 8, 't': 6, 'm': 0, 'y': 9, 'o': 7})
17 3712 + 463 = 4179 (mapping: {'d': 2, 'n': 1, 'e': 6, 's': 7, 't': 3, 'm': 0, 'y': 9, 'o': 4})
18 8432 + 914 = 9346 (mapping: {'d': 2, 'n': 3, 'e': 1, 's': 4, 't': 8, 'm': 0, 'y': 6, 'o': 9})
19 5732 + 647 = 6379 (mapping: {'d': 2, 'n': 3, 'e': 4, 's': 7, 't': 5, 'm': 0, 'y': 9, 'o': 6})
20 8542 + 915 = 9457 (mapping: {'d': 2, 'n': 4, 'e': 1, 's': 5, 't': 8, 'm': 0, 'y': 7, 'o': 9})
21 7643 + 826 = 8469 (mapping: {'d': 3, 'n': 4, 'e': 2, 's': 6, 't': 7, 'm': 0, 'y': 9, 'o': 8})
22 6853 + 728 = 7581 (mapping: {'d': 3, 'n': 5, 'e': 2, 's': 8, 't': 6, 'm': 0, 'y': 1, 'o': 7})
23 8324 + 913 = 9237 (mapping: {'d': 4, 'n': 2, 'e': 1, 's': 3, 't': 8, 'm': 0, 'y': 7, 'o': 9})
24 6524 + 735 = 7259 (mapping: {'d': 4, 'n': 2, 'e': 3, 's': 5, 't': 6, 'm': 0, 'y': 9, 'o': 7})
25 7534 + 825 = 8359 (mapping: {'d': 4, 'n': 3, 'e': 2, 's': 5, 't': 7, 'm': 0, 'y': 9, 'o': 8})
26 6415 + 734 = 7149 (mapping: {'d': 5, 'n': 1, 'e': 3, 's': 4, 't': 6, 'm': 0, 'y': 9, 'o': 7})
27 7316 + 823 = 8139 (mapping: {'d': 6, 'n': 1, 'e': 2, 's': 3, 't': 7, 'm': 0, 'y': 9, 'o': 8})
28 2817 + 364 = 3185 (mapping: {'d': 7, 'n': 1, 'e': 6, 's': 0, 't': 2, 'm': 0, 'y': 5, 'o': 3})
29 9567 + 1885 = 10652 (mapping: {'d': 7, 'n': 6, 'e': 8, 's': 5, 't': 9, 'm': 1, 'y': 2, 'o': 0})
30 6419 + 724 = 7143 (mapping: {'d': 9, 'n': 1, 'e': 2, 's': 4, 't': 6, 'm': 0, 'y': 3, 'o': 7})
31 3719 + 457 = 4176 (mapping: {'d': 9, 'n': 1, 'e': 5, 's': 7, 't': 3, 'm': 0, 'y': 6, 'o': 4})
32 2819 + 364 = 3187 (mapping: {'d': 9, 'n': 1, 'e': 6, 's': 0, 't': 2, 'm': 0, 'y': 7, 'o': 3})
33 7429 + 814 = 8243 (mapping: {'d': 9, 'n': 2, 'e': 1, 's': 4, 't': 7, 'm': 0, 'y': 3, 'o': 8})
34 3829 + 458 = 4287 (mapping: {'d': 9, 'n': 2, 'e': 5, 's': 8, 't': 3, 'm': 0, 'y': 7, 'o': 4})
35 7539 + 815 = 8354 (mapping: {'d': 9, 'n': 3, 'e': 1, 's': 5, 't': 7, 'm': 0, 'y': 4, 'o': 8})
36 7649 + 816 = 8465 (mapping: {'d': 9, 'n': 4, 'e': 1, 's': 6, 't': 7, 'm': 0, 'y': 5, 'o': 8})
37 5849 + 638 = 6487 (mapping: {'d': 9, 'n': 4, 'e': 3, 's': 8, 't': 5, 'm': 0, 'y': 7, 'o': 6})
38 Process exited with code: 0
```

RESULT: Hence, the implementation of CSP is done successfully.

Aim:- To implement CSP i.e. Cryptarithmic problem $SEND + MORE = MONEY$.

Problem Formulation :-

Given an expression where two words add 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 :-

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

Goal State :-

In the goal state the number 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 $E=5$.

$$\begin{array}{r} E \\ + 0 \\ \hline N \end{array}$$

As 0 is zero, there is some carry factor $E \neq N$ can't be same.

$$\begin{array}{r} C_2 \\ E \\ + 0 \\ \hline N \end{array} \qquad \begin{array}{r} 1 \\ 5 \\ + 0 \\ \hline 6 \end{array}$$

$$\therefore N=6$$

- Moving ahead,

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

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

On adding last 2 terms,

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

Final state -

$0c_3$ $1c_2$ $1c_1$
 $\boxed{9}_s$ $\boxed{5}_t$ $\boxed{6}_N$ $\boxed{7}_D$

$\boxed{1}_M$ $\boxed{0}_O$ $\boxed{8}_R$ $\boxed{5}_E$

$\boxed{1}_M$ $\boxed{0}_O$ $\boxed{6}_N$ $\boxed{5}_E$ $\boxed{2}_Y$