Manas Jha
RA1911003010643

Date

LAB-3

Aim: To implement the constraint Satisfaction Problem based on the given constraints.

SEND + MORE = MONEY

```
  5  4  3  2  1
     S  E  N  D
     M  O  R  E
  ------------
+    C3 C2 C1
  ------------
  M  O  N  E  Y
```

```
   9  5  6  7
+  1  0  8  5
  ------------
  10  6  5  2
```

S	9
E	5
N	6
D	7

M	1
O	0
R	8
E	5

Algorithm:

→ From column 5, M=1, since it is only carry-over possible from sum of 2 digits number in column 4.

→ To produce a carry from column 4 to column 5 'S+M' is atleast 9 so 'S=8 or 9' so 'S+M=9 or 10' & so 'O=0 or 1'. But M=1', so 'O=0'

→ If there is a carry from column 3 to 4 then 'E=9' & 'N=0'. But 'O=0' so there is no carry & 'S=9' & 'C3=0'.

Date

→ If there is no carry from column 2 to 3 then 'E = N' which is impossible. therefore there is carry & 'N = E+1' & 'C2 = 1'

→ If there is a carry from column 1 to 2 then 'N + R = E mod 10' & 'N = E+1' so 'E+1+R = E mod 10' so 'R = 9' but 'S = 9', so there must be carry from column 1 to 2. Therefore 'C1 = 1' & 'R = 8'.

→ To produce carry 'C1 = 1' from column 1 to 2, we must have 'D+E = 10+Y'

→ If E were 6 & D+E atleast 12 then D would be 7, but 'N = E+1' & N would also be 7 which is impossible. ∴ 'E = 5' & 'N = 6'.

→ D+E is atleast 12 for that we get 'D = 7' & 'Y = 2'.

Manas Jha

RA1911003010643

```
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))

print('EAT + THAT = APPLE ')

solve2('POINT + ZERO = ENERGY ')
```