Expt 3:- Constraint Specification Problem.

Problem:- SEND + MORE = "MONEY"

*(Solve the following puzzle by assigning numeral (0-9) in such a way that each letter is assigned unique digit which satisfy addition). (Problem Formulation)

Constraint:- No two letters have the same value. (The constraints of arithmetic).

```
  S E N D
+ M O R E
---------
M O N E Y
```

Initial State:-

- S = ?; E = ?; N = ?; D = ?; M = ?; O = ?; N = ?; E = ?, Y = ?

- C4 C3 C2 C1 ← Carry

```
      S E N D
(+)   M O R E
    M O N E Y
```

Algorithm:-

```
5  4  3  2  1
   S  E  N  D

+  M  O  R  E

   C3 C2 C1
_______________
M  O  N  E  Y
_______________
```

(1.) From column 5, M=1, since it is only carry over possible from sum of 2 single digit number in column 4.

(2.) To produce a carry from column: 'S+M' is at least 9 so 'S=8 or 9' so 'S+M=9 or 10' & so 'O=0 or 1'. But 'M=1', so 'O=0'.

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

(4.) If there is no carry from column 2 to 3 then 'E=N' which is impossible, therefor there is carry & 'N=E+1' & 'E2=1'.

(5.) If there is carry from column 1 to 2 then 'N = k+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'.

Solution:-

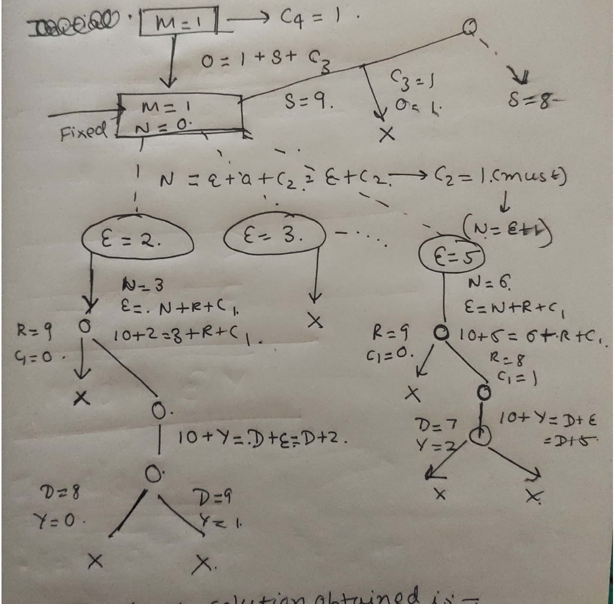
Constraints $Y = D + E \rightarrow C_1.$

$E = N + R + C_1 \rightarrow C_2.$

$N = E + O + C_2 \rightarrow C_3.$

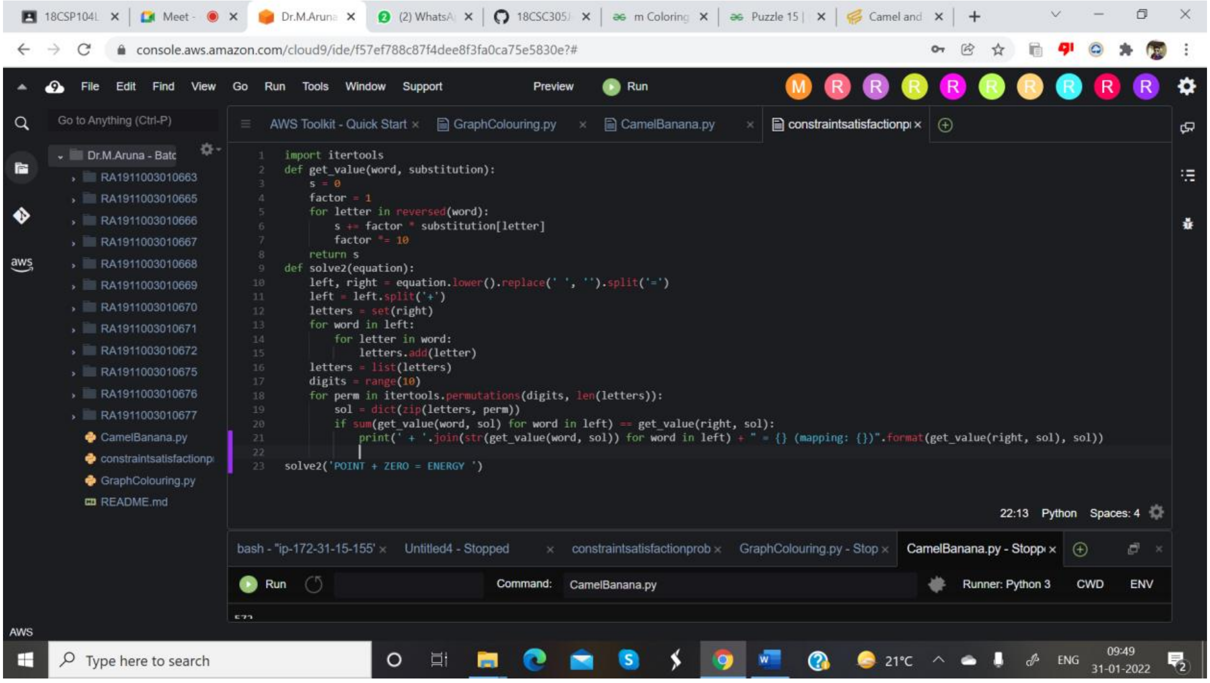
$O = S + M + C_3 \rightarrow C_4.$

$M = C_4.$



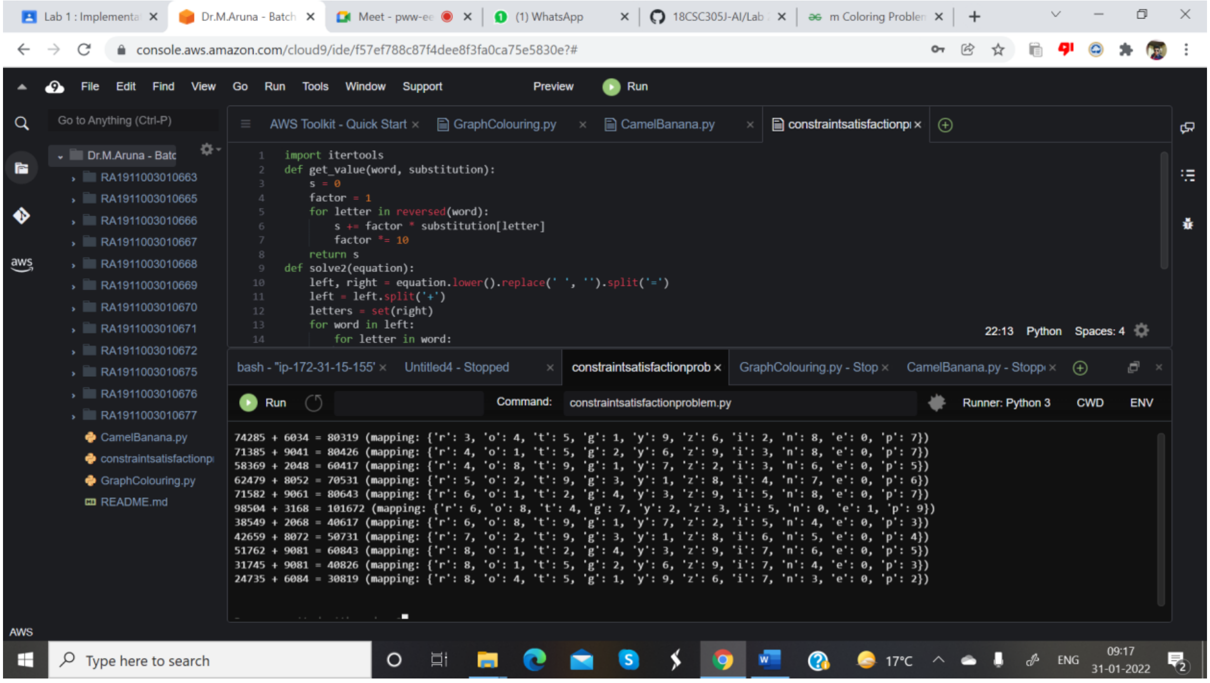
The first solution obtained is:-

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



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



```
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:
```

Command: constraintsatisfactionproblem.py

```
74285 + 6034 = 80319 (mapping: {'r': 3, 'o': 4, 't': 5, 'g': 1, 'y': 9, 'z': 6, 'i': 2, 'n': 8, 'e': 0, 'p': 7})
71385 + 9041 = 80426 (mapping: {'r': 4, 'o': 1, 't': 5, 'g': 2, 'y': 6, 'z': 9, 'i': 3, 'n': 8, 'e': 0, 'p': 7})
58369 + 2048 = 60417 (mapping: {'r': 4, 'o': 8, 't': 9, 'g': 1, 'y': 7, 'z': 2, 'i': 3, 'n': 6, 'e': 0, 'p': 5})
62479 + 8052 = 70531 (mapping: {'r': 5, 'o': 2, 't': 9, 'g': 3, 'y': 1, 'z': 8, 'i': 4, 'n': 7, 'e': 0, 'p': 6})
71582 + 9061 = 80643 (mapping: {'r': 6, 'o': 1, 't': 2, 'g': 4, 'y': 3, 'z': 9, 'i': 5, 'n': 8, 'e': 0, 'p': 7})
98504 + 3168 = 101672 (mapping: {'r': 6, 'o': 8, 't': 4, 'g': 7, 'y': 2, 'z': 3, 'i': 5, 'n': 0, 'e': 1, 'p': 9})
38549 + 2068 = 40617 (mapping: {'r': 6, 'o': 8, 't': 9, 'g': 1, 'y': 7, 'z': 2, 'i': 5, 'n': 4, 'e': 0, 'p': 3})
42659 + 8072 = 50731 (mapping: {'r': 7, 'o': 2, 't': 9, 'g': 3, 'y': 1, 'z': 8, 'i': 6, 'n': 5, 'e': 0, 'p': 4})
51762 + 9081 = 60843 (mapping: {'r': 8, 'o': 1, 't': 2, 'g': 4, 'y': 3, 'z': 9, 'i': 7, 'n': 6, 'e': 0, 'p': 5})
31745 + 9081 = 40826 (mapping: {'r': 8, 'o': 1, 't': 5, 'g': 2, 'y': 6, 'z': 9, 'i': 7, 'n': 4, 'e': 0, 'p': 3})
24735 + 6084 = 30819 (mapping: {'r': 8, 'o': 4, 't': 5, 'g': 1, 'y': 9, 'z': 6, 'i': 7, 'n': 3, 'e': 0, 'p': 2})
```

Result:-

The constraint satisfaction problem(CRYPTARITHMETIC Problem)was successfully implemented using python