Artificial Intelligence

Week 3

Topic:

Constraint Satisfaction Problem

AIM:

To implement a cryptarithmetic problem

Problem Statement :

Cryptarithmetic Problem is a type of constraint satisfaction problem where the we have to assign digits to alphabets or other symbols by using some constraints.Constrains are listed below

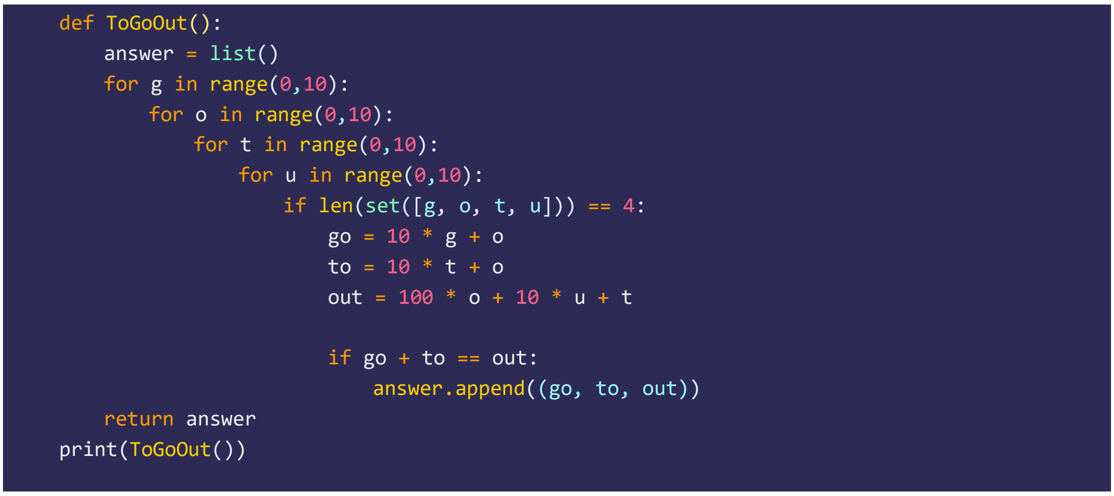
The rules or constraints on a cryptarithmetic problem are as follows:

- There should be a unique digit to be replaced with a unique alphabet.
- The result should satisfy the predefined arithmetic rules, i.e., 2+2 =4, nothing else.
- Digits should be from **0-9** only.
- There should be only one carry forward, while performing the addition operation on a problem.
- The problem can be solved from both sides, i.e., lefthand side (L.H.S), or righthand side (R.H.S)

Algorithm :

1) By brute force technique we are assigning the numbers between 0 to 9 to each character .
2) Now check whether each chacter is assigned to different numbers or not by
3) Inserting all the unique codes to set and then check whether the size of the set is 4 0r not.
4) If the set size is 4 then we are checking whether To + go == out.
5) We can create a list which keep track the possible answers .If the step3 is satified simply add the four numbers to our answer.
6) Return the answer which contains all possible solution to our problem.

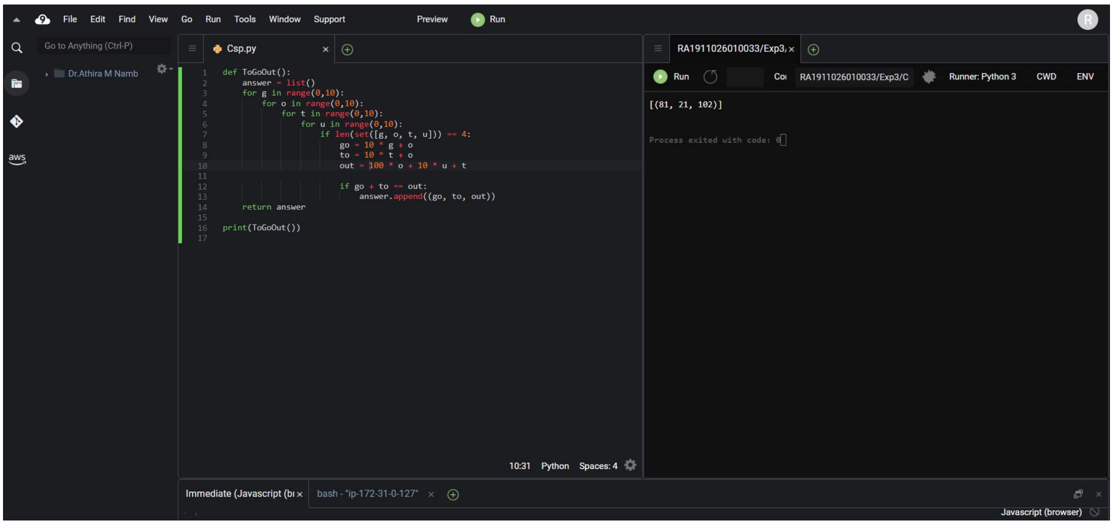
Code :



```python
def ToGoOut():
    answer = list()
    for g in range(0,10):
        for o in range(0,10):
            for t in range(0,10):
                for u in range(0,10):
                    if len(set([g, o, t, u])) == 4:
                        go = 10 * g + o
                        to = 10 * t + o
                        out = 100 * o + 10 * u + t

                        if go + to == out:
                            answer.append((go, to, out))
    return answer
print(ToGoOut())
```

Output:



```python
def ToGoOut():
    answer = list()
    for g in range(0,10):
        for o in range(0,10):
            for t in range(0,10):
                for u in range(0,10):
                    if len(set([g, o, t, u])) == 4:
                        go = 10 * g + o
                        to = 10 * t + o
                        out = 100 * o + 10 * u + t

                        if go + to == out:
                            answer.append((go, to, out))
    return answer

print(ToGoOut())
```

```
[(81, 21, 102)]

Process exited with code: 0
```

Observation

For “O” There are two possibilities, either it can be 0 or 1.It cannot be zero.since “O” is a leading character,i.e. the first digit in a number.

Time complexity is O(9 * 9 * 9 * 9)
Since we are running a 4 for loops

Space Complexity is O(n)
Here n is the number of solutions

Result :

the CryptArithmetic problem(TO + GO = OUT) which is a CSP is solved.