

## **Exp 3 Constrain satisfaction problem**

### **TEAM – AUTOMATA LAB**

RA1911031010128-VEER VISWAJEET SWAMI

RA1911031010129-VIJAY RENGARAJ R

RA1911031010130-THINAKAR R

RA1911031010131-NAMARATA MISHRA

RA1911031010138-HUSNA QASIM

#### **Problem chosen:**

CRYPT ARETHEMTIC PROBLEM

#### **Problem statement:**

Cryptarithmic Problem is a type of constraint satisfaction problem where the game is about digits and its unique replacement either with alphabets or other symbols. In cryptarithmic problem, the digits (0-9) get substituted by some possible alphabets or symbols. The task in cryptarithmic problem is to substitute each digit with an alphabet to get the result arithmetically correct.

We can perform all the arithmetic operations on a given cryptarithmic problem.

#### **Code:**

```
import itertools
```

```
def get_value(word, sub):
```

```
    s = 0
```

```
    factor = 1
```

```
for letter in reversed(word):
```

```
    s += factor * sub[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__':
```

```
first_word = input().strip()
```

```
second_word = input().strip()
```

```
ans = input().strip()
```

```
solve2('%s + %s = %s'%(first_word, second_word, ans))
```

## OUTPUT:

```
In [1]: import itertools

def get_value(word, sub):
    s = 0
    factor = 1
    for letter in reversed(word):
        s += factor * sub[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__':
    first_word = input().strip()
    second_word = input().strip()
    ans = input().strip()
    solve2('%s + %s = %s'%(first_word, second_word, ans))
```

base  
ball  
games  
2461 + 2455 = 4916 (mapping: {'m': 9, 's': 6, 'b': 2, 'a': 4, 'e': 1, 'l': 5, 'g': 0})  
2483 + 2455 = 4938 (mapping: {'m': 9, 's': 8, 'b': 2, 'a': 4, 'e': 3, 'l': 5, 'g': 0})  
7483 + 7455 = 14938 (mapping: {'m': 9, 's': 8, 'b': 7, 'a': 4, 'e': 3, 'l': 5, 'g': 1})

## Result:

The problem statement is solved using permutations and combinations (using python language in JUPYTER environment).