

EX3: Constraint satisfaction Problems

Name: Reeti Jha

Reg No: RA1911030010121

AIM: To implement the Constraint Satisfaction Problem based on the given constraints

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))
```

```
print('EAT + THAT = APPLE ')
```

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



```
1 import itertools
2
3 def get_value(word, substitution):
4     s = 0
5     factor = 1
6     for letter in reversed(word):
7         s += factor * substitution[letter]
8         factor *= 10
9     return s
10
11 def solve2(equation):
12     left, right = equation.lower().replace(' ', '').split('=')
13     left = left.split('+')
14     letters = set(right)
15     for word in left:
16         for letter in word:
17             letters.add(letter)
18     letters = list(letters)
19
20     digits = range(10)
21     for perm in itertools.permutations(digits, len(letters)):
22         sol = dict(zip(letters, perm))
23
24         if sum(get_value(word, sol) for word in left) == get_value(right, sol):
25             print(' + '.join(str(get_value(word, sol)) for word in left) + " = {} (mapping: {})"
26                   .format(get_value(right, sol), sol))
27
28 print('EAT + THAT = APPLE ')
29 solve2('POINT + ZERO = ENERGY ')
```

Output:

```
121.py x +
121/121.py - Stopped x +
Run Command: 121/121.py Runner:
EAT + THAT = APPLE
71582 + 9061 = 80643 (mapping: {'t': 2, 'i': 5, 'g': 4, 'p': 7, 'y': 3, 'n': 8, 'o': 1, 'z': 9, 'e': 0, 'r': 6})
51762 + 9081 = 60843 (mapping: {'t': 2, 'i': 7, 'g': 4, 'p': 5, 'y': 3, 'n': 6, 'o': 1, 'z': 9, 'e': 0, 'r': 8})
98504 + 3168 = 101672 (mapping: {'t': 4, 'i': 5, 'g': 7, 'p': 9, 'y': 2, 'n': 0, 'o': 8, 'z': 3, 'e': 1, 'r': 6})
74285 + 6034 = 80319 (mapping: {'t': 5, 'i': 2, 'g': 1, 'p': 7, 'y': 9, 'n': 8, 'o': 4, 'z': 6, 'e': 0, 'r': 3})
71385 + 9041 = 80426 (mapping: {'t': 5, 'i': 3, 'g': 2, 'p': 7, 'y': 6, 'n': 8, 'o': 1, 'z': 9, 'e': 0, 'r': 4})
24735 + 6084 = 30819 (mapping: {'t': 5, 'i': 7, 'g': 1, 'p': 2, 'y': 9, 'n': 3, 'o': 4, 'z': 6, 'e': 0, 'r': 8})
31745 + 9081 = 40826 (mapping: {'t': 5, 'i': 7, 'g': 2, 'p': 3, 'y': 6, 'n': 4, 'o': 1, 'z': 9, 'e': 0, 'r': 8})
58369 + 2048 = 60417 (mapping: {'t': 9, 'i': 3, 'g': 1, 'p': 5, 'y': 7, 'n': 6, 'o': 8, 'z': 2, 'e': 0, 'r': 4})
62479 + 8052 = 70531 (mapping: {'t': 9, 'i': 4, 'g': 3, 'p': 6, 'y': 1, 'n': 7, 'o': 2, 'z': 8, 'e': 0, 'r': 5})
38549 + 2068 = 40617 (mapping: {'t': 9, 'i': 5, 'g': 1, 'p': 3, 'y': 7, 'n': 4, 'o': 8, 'z': 2, 'e': 0, 'r': 6})
42659 + 8072 = 50731 (mapping: {'t': 9, 'i': 6, 'g': 3, 'p': 4, 'y': 1, 'n': 5, 'o': 2, 'z': 8, 'e': 0, 'r': 7})

Process exited with code: 0
```

Result: Hence we have successfully studied constraint satisfaction problems.