EXPERIMENT NO: 3 **A N V SREEVISHNU**
DATE: 12/02/2021 **RA1811003010333**

IMPLEMENTATION OF CONSTRAINT SATISFACTION PROBLEM

Aim: To implement Constraint Satisfaction Problem (Cryptarithmetic Problem) in AI using Python.

Procedure/Algorithm:

- First, create a list of all the characters that need assigning to pass to

  solve.
- If all characters are assigned, return true if the puzzle is solved, false otherwise.
- Otherwise, consider the first unassigned character for (every

  possible choice among the digits not in use) make that choice

  and then if

  recursion successful, return true
- If not successful, unmake assignment and try another digitively try to

  assign the r est of the characters.
- If all digits have been tried and nothing worked, return false .

Code:

```
from re import
sub def solve(q):
try:
    n = (i for i in q if i.isalpha()).__next__()
except StopIteration:
```

```
        return q if eval(sub(r'(^|[^0-9])0+([1-9]+)', r'\1\2',          q)) else
False       else:
        for i in (str(i) for i in range(10) if str(i) not in q):
            res = solve(q.replace(n, str(i)))
if res:
                return res
return False if __name__
== "__main__":
    query = str(input("Enter the
String:"))       r = solve(query)       if r:
        print(r)          for j in
range(len(query)):
print(query[j], "-->", r[j])
else:
        print("Solution Not Found")
```

Output:

```
Enter the String:THREE + THREE + ONE == SEVEN
23577 + 23577 + 817 == 47971
T --> 2
H --> 3
R --> 5
E --> 7
E --> 7
  -->
+ --> +
  -->
T --> 2
H --> 3
R --> 5
E --> 7
E --> 7
  -->
+ --> +
  -->
O --> 8
N --> 1
E --> 7
  -->
= --> =
= --> =
  -->
S --> 4
E --> 7
V --> 9
E --> 7
N --> 1
```

Result: Thus,the implementation of Constraint Satisfaction Problem (Cryptarithmetic Problem) in AI using Python has been successfully done.