**Exp No:** 3 **Implementation of Constrain-Satisfactory Problem**.(Cryptarithmetics)

**Date:** 07-02-2023

**AIM:**

To write a python program for cryptarithmetic using permutation method.

**CODE:**

import itertools

def solve2():

letters = ('b', 'a', 's', 'e', 'l', 'g', 'm')

digits = range(10)

for perm in itertools.permutations(digits, len(letters)):

sol = dict(zip(letters, perm))

if sol['b'] == 0 or sol['g'] == 0:

continue

base = 1000 \* sol['b'] + 100 \* sol['a'] + 10 \* sol['s'] + sol['e']

ball = 1000 \* sol['b'] + 100 \* sol['a'] + 10 \* sol['l'] + sol['l']

games = 10000 \* sol['g'] + 1000 \* sol['a'] + 100 \* sol['m'] + 10 \* sol['e'] + sol['s']

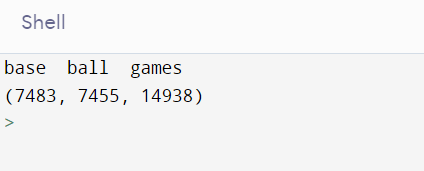
if base + ball == games:

print(" base"," ball"," games")

return base,ball,games

print(solve2())

**Sample Input and Output:**



**Result:**

The crypt-arithmetic problem is solved using permutations and steps are executed in order and been successfully implemented.