PROGRAM

from ortools.sat.python import cp\_model

model = cp\_model.CpModel()

base = 10

c = model.NewIntVar(1, 9, 'C')

p = model.NewIntVar(0, 9, 'P')

i = model.NewIntVar(1, 9, 'I')

s = model.NewIntVar(0, 9, 'S')

f = model.NewIntVar(1, 9, 'F')

u = model.NewIntVar(0, 9, 'U')

n = model.NewIntVar(0, 9, 'N')

t = model.NewIntVar(1, 9, 'T')

r = model.NewIntVar(0, 9, 'R')

e = model.NewIntVar(0, 9, 'E')

# List of variables

letters = [c, p, i, s, f, u, n, t, r, e]

model.AddAllDifferent(letters)

# CP + IS + FUN = TRUE

model.Add(c \* 10 + p + i \* 10 + s + f \* base\*\*2 + u \* 10 + n == t \* 10\*\*3 + r \* 10\*\*2 + u \* 10 + e)

class VarArraySolutionPrinter(cp\_model.CpSolverSolutionCallback):

def \_\_init\_\_(self, variables):

cp\_model.CpSolverSolutionCallback.\_\_init\_\_(self)

self.\_\_variables = variables

self.\_\_solution\_count = 0

def on\_solution\_callback(self):

self.\_\_solution\_count += 1

for v in self.\_\_variables:

print('%s=%i ' %(v, self.Value(v)), end='')

print()

def solution\_count(self):

return self.\_\_solution\_count

solver = cp\_model.CpSolver()

solution\_printer = VarArraySolutionPrinter(letters)

status = solver.SearchForAllSolutions(model, solution\_printer)

OUTPUT

