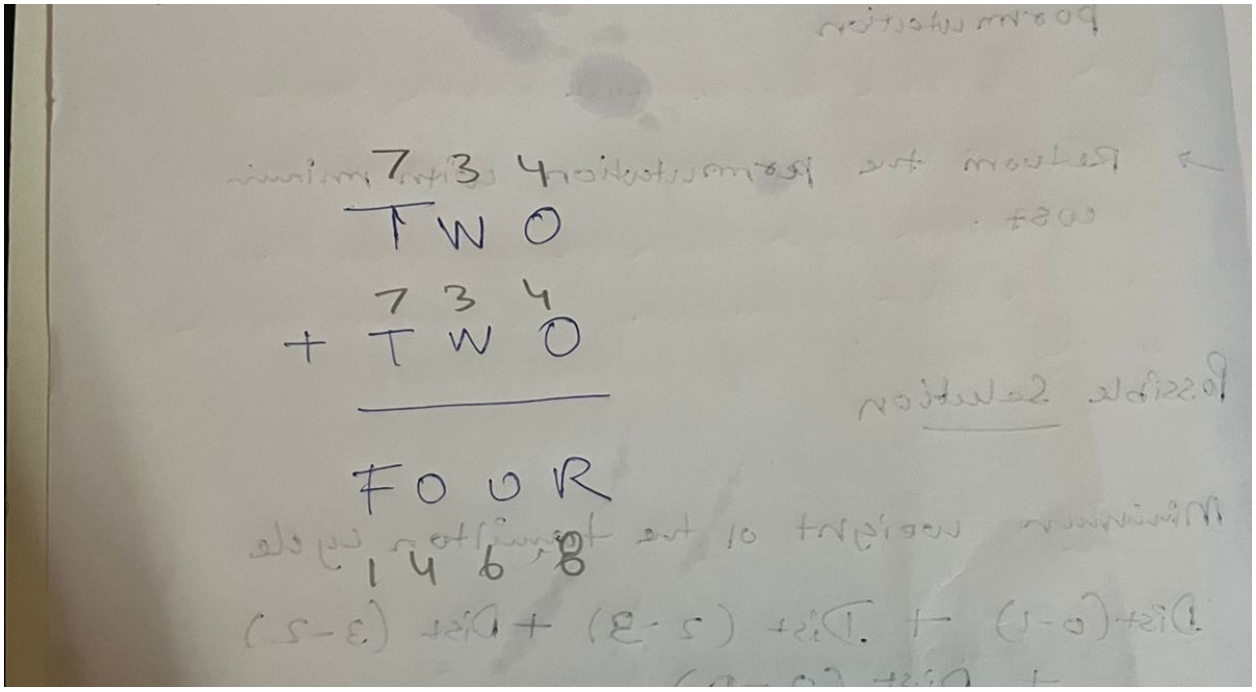
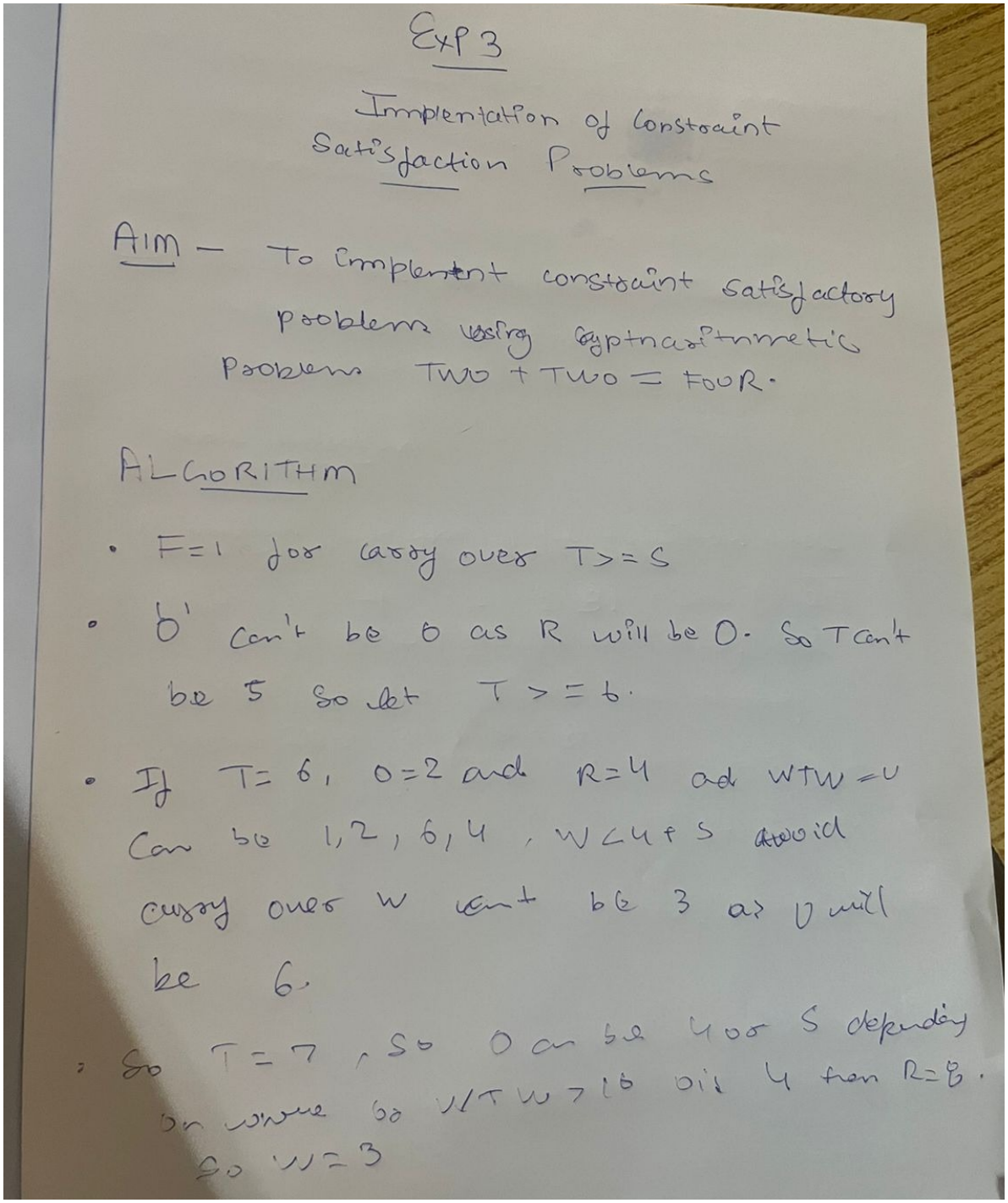
EX3
Implementation of constraint satisfaction problems

Harsh Saxena
RA1911033010060





Exp 3

Implementation of Constraint Satisfaction Problems

Aim – To implement constraint satisfactory problems using cryptarithmetic Problem TWO + TWO = FOUR.

ALGORITHM

- F=1 for carry over T>=5
- O' can't be 0 as R will be 0. So T can't be 5 so let T >= 6.
- If T= 6, O=2 and R=4 and W+W=U Can be 1, 2, 6, 4, W<4+5 to avoid carry over W can't be 3 as U will be 6.
- So T=7, so O can be 4 or 5 depending on whether so W+W>10 giving 4 from R=8. So W=3

CODE

```
from re import sub
```

```
def solve(q):
  try:
    n = next(i for i in q if i.isalpha())
  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):
      r = solve(q.replace(n,str(i)))
      if r:
        return r
    return False

if __name__ == "__main__":
  query = "TWO + TWO == FOUR"
  r = solve(query)
  print(query)
  print(r)
```