AI LAB MANUAL

Exp 3: Constraint Satisfaction Problem

Name:	SHAMUNESH P
Reg.no:	RA1911030010122
Problem:	Constraint Satisfaction Problem
Date:	24-01-22

Constraint Satisfaction Problem:

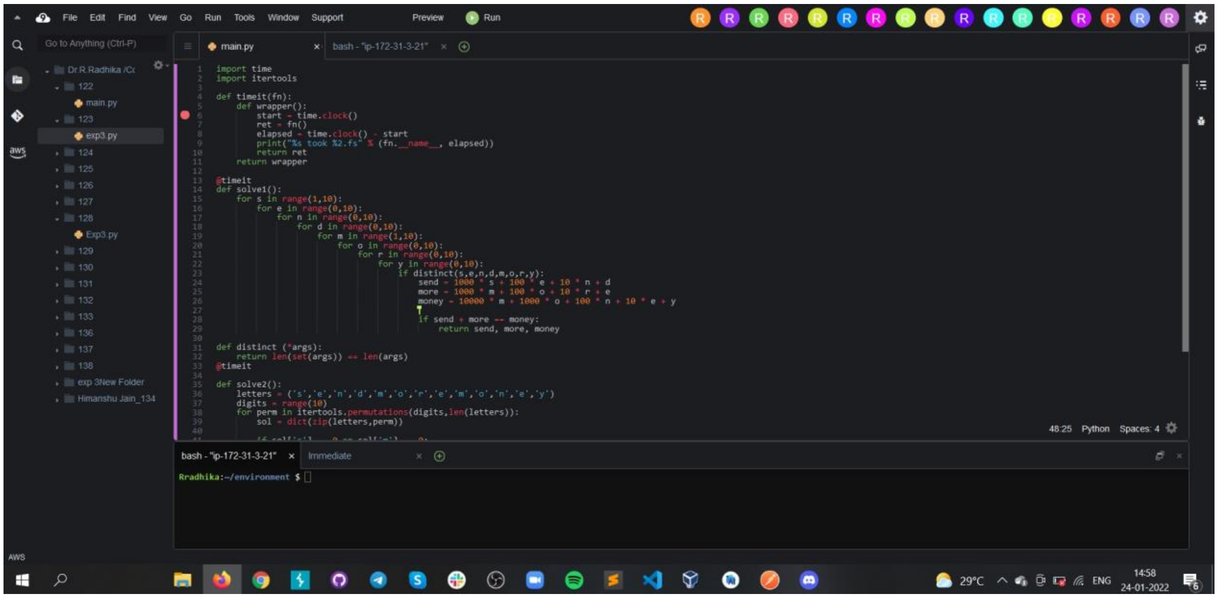
Constraint satisfaction problems are mathematical questions defined as a set of objects whose state must satisfy a number of constraints or limitations. CSPs represent the entities in a problem as a homogeneous collection of finite constraints over variables, which is solved by constraint satisfaction methods.

Code: (Python)

```
def solutions():
    all_solutions = list()
    for s in range(9, -1, -1):
        for e in range(9, -1, -1):
            for n in range(9, -1, -1):
                for d in range(9, -1, -1):
                    for m in range(9, 0, -1):
                        for o in range(9, -1, -1):
                            for r in range(9, -1, -1):
                                for y in range(9, -1, -1):
                            if len(set([s, e, n, d, m, o, r, y])) == 8:
                            send = 1000 * s + 100 * e + 10 * n + d
                            more = 1000 * m + 100 * o + 10 * r + e
                            money = 10000 * m + 1000 * o + 100 * n + 10 * e + y
                    if send + more == money:
                        all_solutions.append((send, more, money))
        return all_solutions
print(solutions())
```

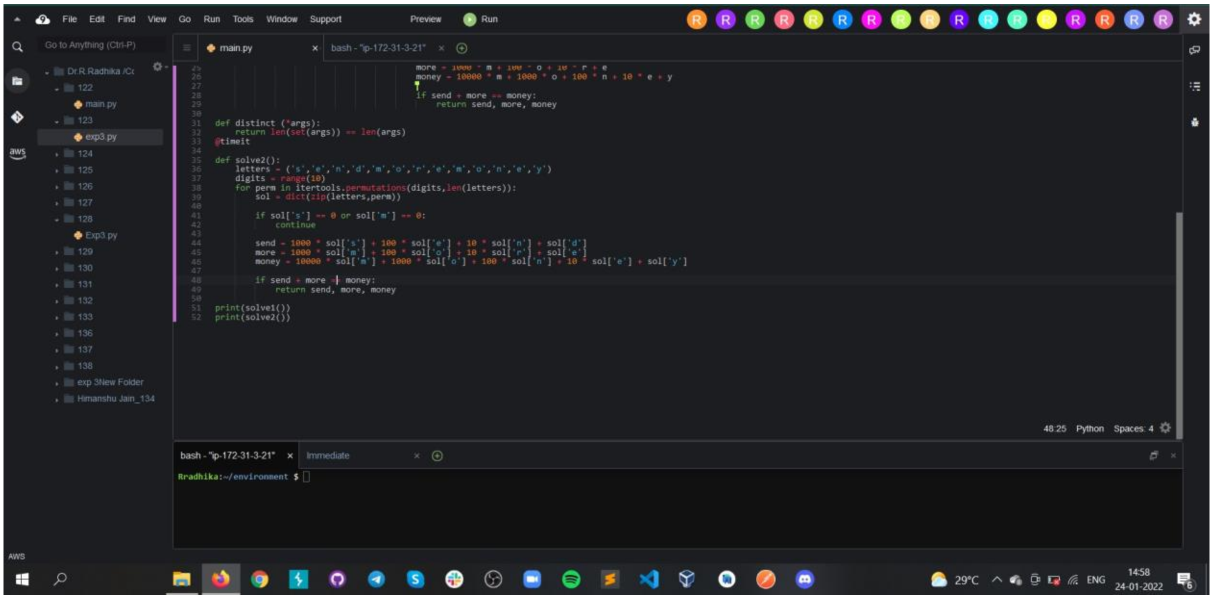
Implementation:

Screenshots



```
import time
import itertools

def timeit(fn):
    def wrapper():
        start = time.clock()
        ret = fn()
        elapsed = time.clock() - start
        print("%s took %2.fs" % (fn.__name__, elapsed))
        return ret
    return wrapper

@timeit
def solve1():
    for s in range(1,10):
        for e in range(0,10):
            for n in range(0,10):
                for d in range(0,10):
                    for m in range(1,10):
                        for o in range(0,10):
                            for r in range(0,10):
                                for y in range(0,10):
                                    if distinct(s,e,n,d,m,o,r,y):
                                        send = 1000 * s + 100 * e + 10 * n + d
                                        more = 1000 * m + 100 * o + 10 * r + e
                                        money = 10000 * m + 1000 * o + 100 * n + 10 * e + y

                                        if send + more == money:
                                            return send, more, money

def distinct (*args):
    return len(set(args)) == len(args)
@timeit

def solve2():
    letters = ('s','e','n','d','m','o','r','e','m','o','n','e','y')
    digits = range(10)
    for perm in itertools.permutations(digits,len(letters)):
        sol = dict(zip(letters,perm))

```

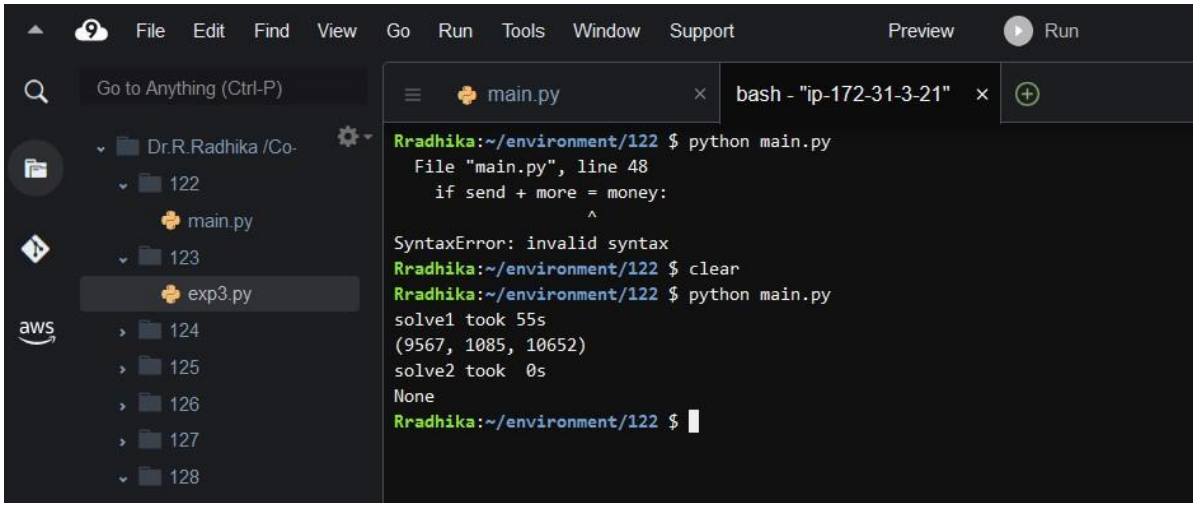
```
Rradhika:~/environment $ 
```



```
                                more = 1000 * m + 100 * o + 10 * r + e
                                money = 10000 * m + 1000 * o + 100 * n + 10 * e + y

                                if send + more == money:
                                    return send, more, money

def distinct (*args):
    return len(set(args)) == len(args)
@timeit

def solve2():
    letters = ('s','e','n','d','m','o','r','e','m','o','n','e','y')
    digits = range(10)
    for perm in itertools.permutations(digits,len(letters)):
        sol = dict(zip(letters,perm))

        if sol['s'] == 0 or sol['m'] == 0:
            continue

        send = 1000 * sol['s'] + 100 * sol['e'] + 10 * sol['n'] + sol['d']
        more = 1000 * sol['m'] + 100 * sol['o'] + 10 * sol['r'] + sol['e']
        money = 10000 * sol['m'] + 1000 * sol['o'] + 100 * sol['n'] + 10 * sol['e'] + sol['y']

        if send + more = money:
            return send, more, money

print(solve1())
print(solve2())
```

```
Rradhika:~/environment $
```

Output:



```
Rradhika:~/environment/122 $ python main.py
  File "main.py", line 48
    if send + more = money:
                   ^
SyntaxError: invalid syntax
Rradhika:~/environment/122 $ clear
Rradhika:~/environment/122 $ python main.py
solve1 took 55s
(9567, 1085, 10652)
solve2 took  0s
None
Rradhika:~/environment/122 $
```