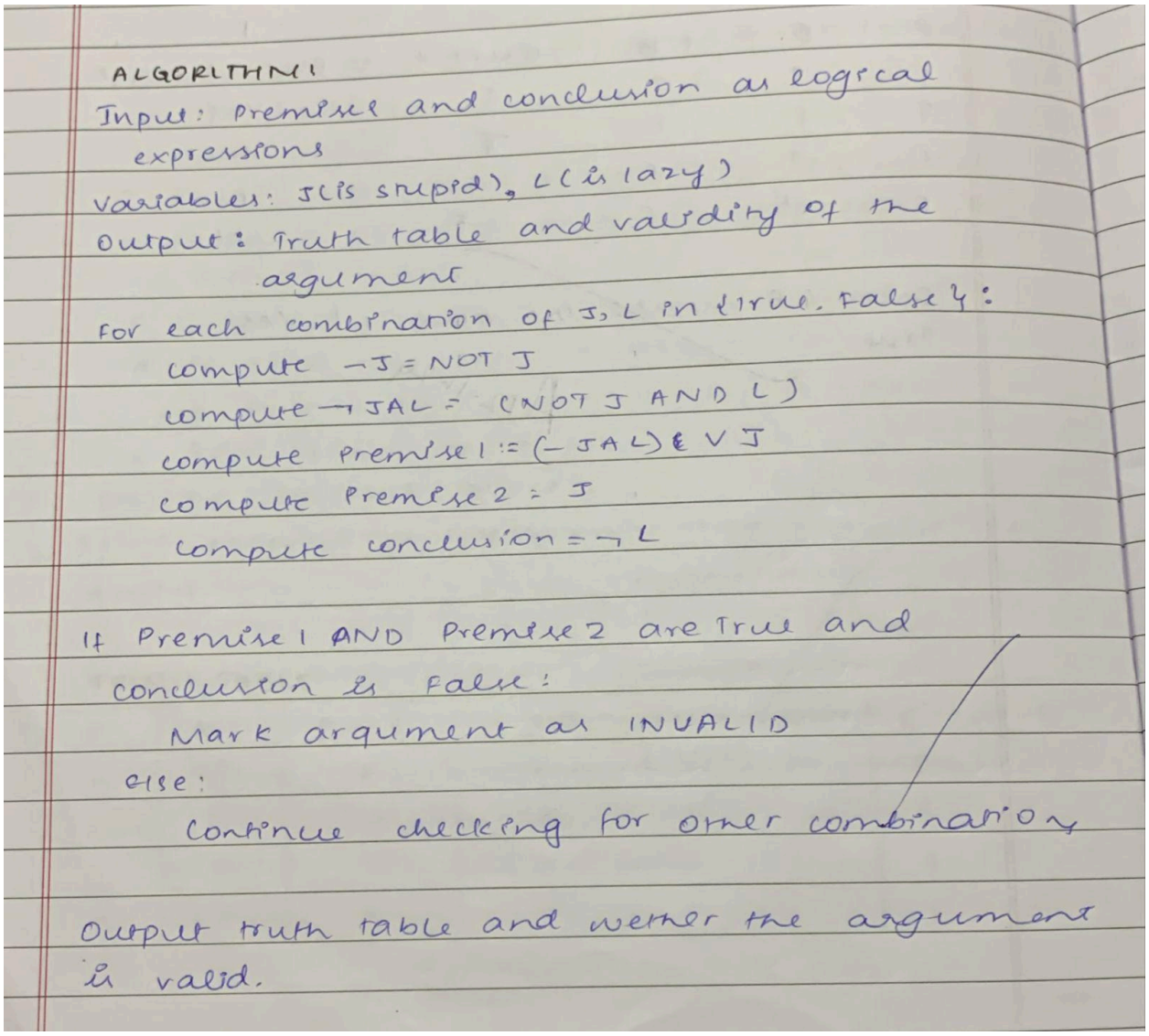
19-11-2024 **WEEK-7**

Preposition Logic

ALGORITHM-



ALGORITHM:
Input: Premises and conclusion as logical expressions
Variables: J(is stupid), L(is lazy)
Output: Truth table and validity of the argument
For each combination of J, L in {True, False}:
compute ¬J = NOT J
compute ¬JAL = (NOT J AND L)
compute Premise1 := (¬JAL) & V J
compute Premise2 : J
compute conclusion = ¬L

If Premise1 AND Premise2 are True and conclusion is False:
Mark argument as INVALID
else:
Continue checking for other combinations

Output truth table and wether the argument is valid.

CODE-

```
import itertools

def truth_table_part1():
    print("Part 1: Truth Table for Argument Validity")
    print("| J | L | ¬J | ¬J∧L | (¬J∧L)∨J | Premise 1 | Premise 2 | ¬L | Conclusion |")
    print("|---|---|----|------|----------|-----------|-----------|----|------------|")

    for J, L in itertools.product([True, False], repeat=2):
        not_J = not J
        not_L = not L
        not_J_and_L = not_J and L
        premise1 = not_J_and_L or J
        premise2 = J
        conclusion = not L
        valid = premise1 and premise2 <= conclusion

        print(f"| {J} | {L} | {not_J} | {not_J_and_L} | {premise1} | {premise1} | {premise2} |
{not_L} | {valid} |")

def ternary_operation(P, Q, R):
    return Q if P else R

def truth_table_part2():
    print("\nPart 2: Truth Table for Ternary Boolean Operation")
    print("| P | Q | R | if P then Q else R |")
    print("|---|---|---|--------------------|")

    for P, Q, R in itertools.product([True, False], repeat=3):
        result = ternary_operation(P, Q, R)
        print(f"| {P} | {Q} | {R} | {result} |")


truth_table_part1()
truth_table_part2()
```

OUTPUT-

Part 1: Truth Table for Argument Validity

J	L	¬J	¬J∧L	(¬J∧L)∨J	Premise 1	Premise 2	¬L	Conclusion
True	True	False	False	True	True	True	False	False
True	False	False	False	True	True	True	True	True
False	True	True	True	True	True	False	False	True
False	False	True	False	False	False	False	True	False

Part 2: Truth Table for Ternary Boolean Operation

P	Q	R	if P then Q else R
True	True	True	True
True	True	False	True
True	False	True	False
True	False	False	False
False	True	True	True
False	True	False	False
False	False	True	True
False	False	False	False