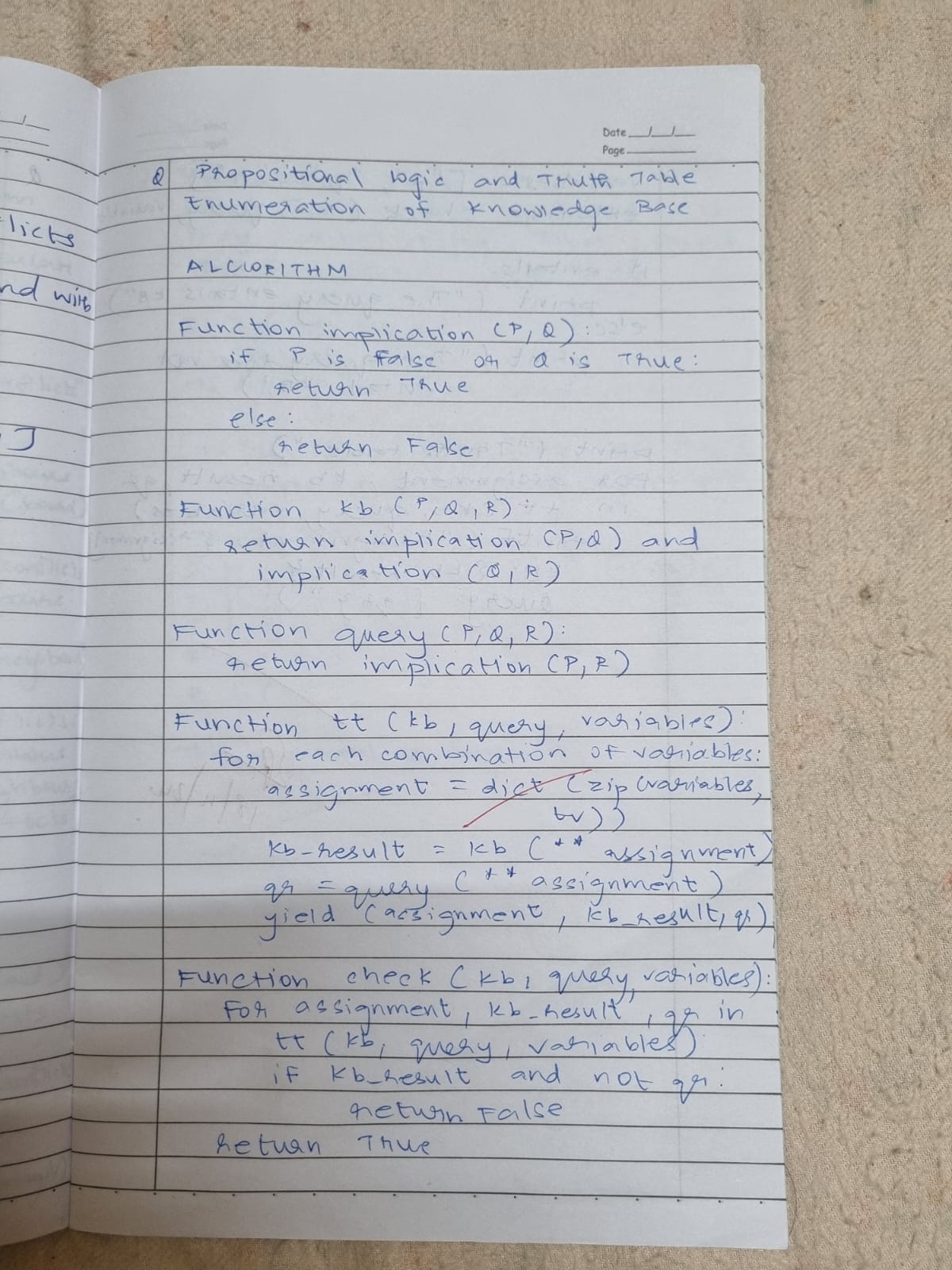
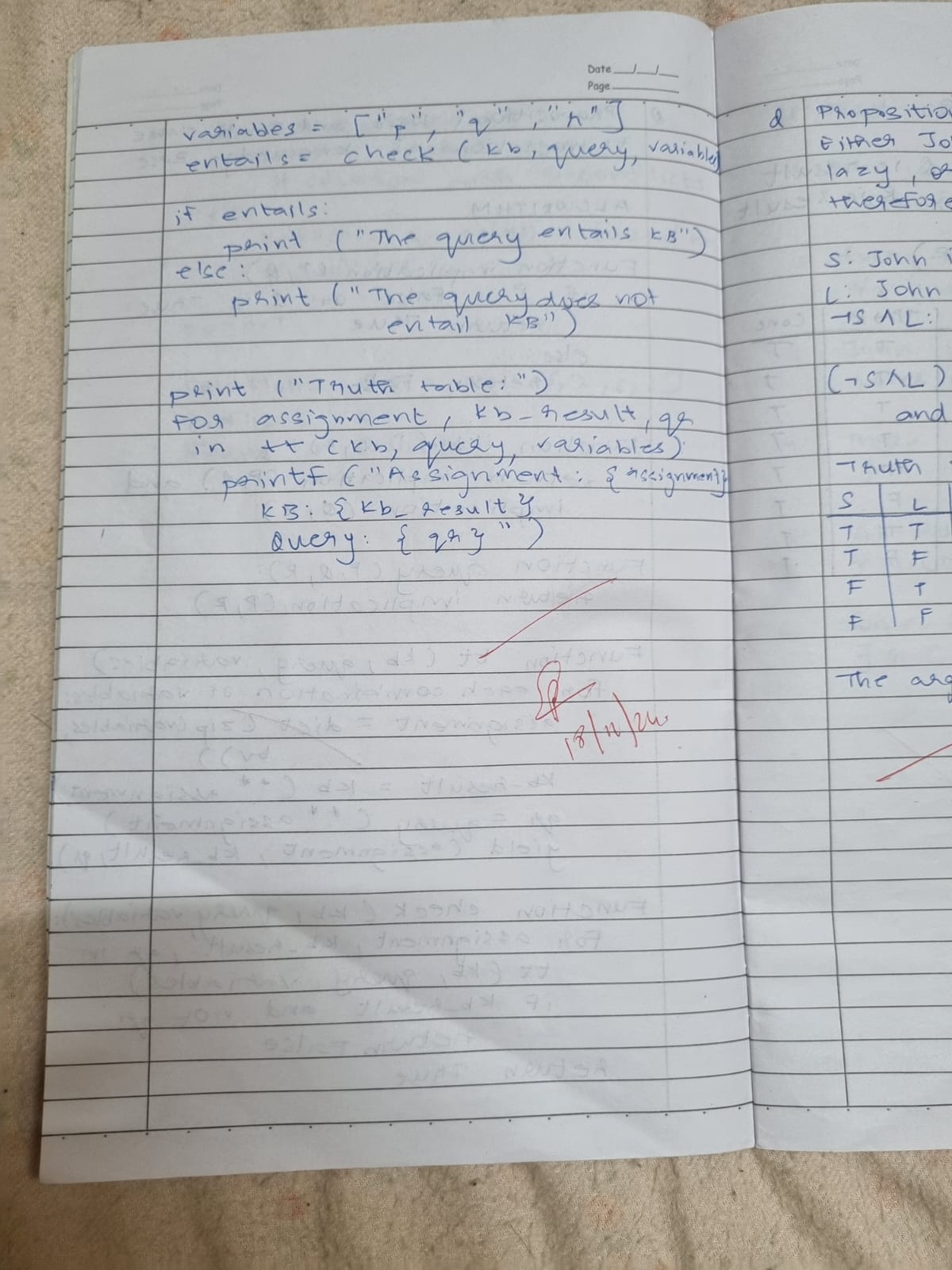
**Prepositional Logic and Truth Table Enumeration**

**Algorithm:**

****

****

**Code:**

from itertools import product

def implication(p, q):

    return not p or q

def kb(p, q, r):

    return implication(p, q) and implication(q, r)

def query(p, q, r):

    return implication(p, r)

def truth\_table(kb, query, variables):

    for truth\_values in product([False, True], repeat=len(variables)):

        # Create a dictionary for the current variable assignments

        assignment = dict(zip(variables, truth\_values))

        kb\_result = kb(\*\*assignment)

        query\_result = query(\*\*assignment)

        yield (assignment, kb\_result, query\_result)

def check\_entailment(kb, query, variables):

    for assignment, kb\_result, query\_result in truth\_table(kb, query, variables):

        if kb\_result and not query\_result:

            return False  # Query does not entail KB

    return True  # Query entails KB

variables = ['p', 'q', 'r']

entails = check\_entailment(kb, query, variables)

if entails:

    print("The query entails the KB.")

else:

    print("The query does NOT entail the KB.")

print("\nTruth Table:")

for assignment, kb\_result, query\_result in truth\_table(kb, query, variables):

    print(f"Assignment: {assignment} | KB: {kb\_result} | Query: {query\_result}")

**Output:**

