**LAB 6:**

**Program:** Implement Prepositional Logic

**Code:**

def PL\_TRUE(model, sentence):

sentence = sentence.replace("A", str(model["A"])).replace("B", str(model["B"])).replace("C", str(model["C"]))

sentence = sentence.replace("and", "and").replace("or", "or").replace("not", "not")

return eval(sentence)

def truth\_table\_entaiment(KB, query, variables):

all\_assignments = []

for i in range(2 \*\* len(variables)):

assignment = {}

for j, var in enumerate(variables):

assignment[var] = (i >> j) & 1

all\_assignments.append(assignment)

for assignment in all\_assignments:

kb\_true = PL\_TRUE(assignment, KB)

query\_true = PL\_TRUE(assignment, query)

if kb\_true and not query\_true:

return False

return True

KB = "(A or C) and (B or not C)"

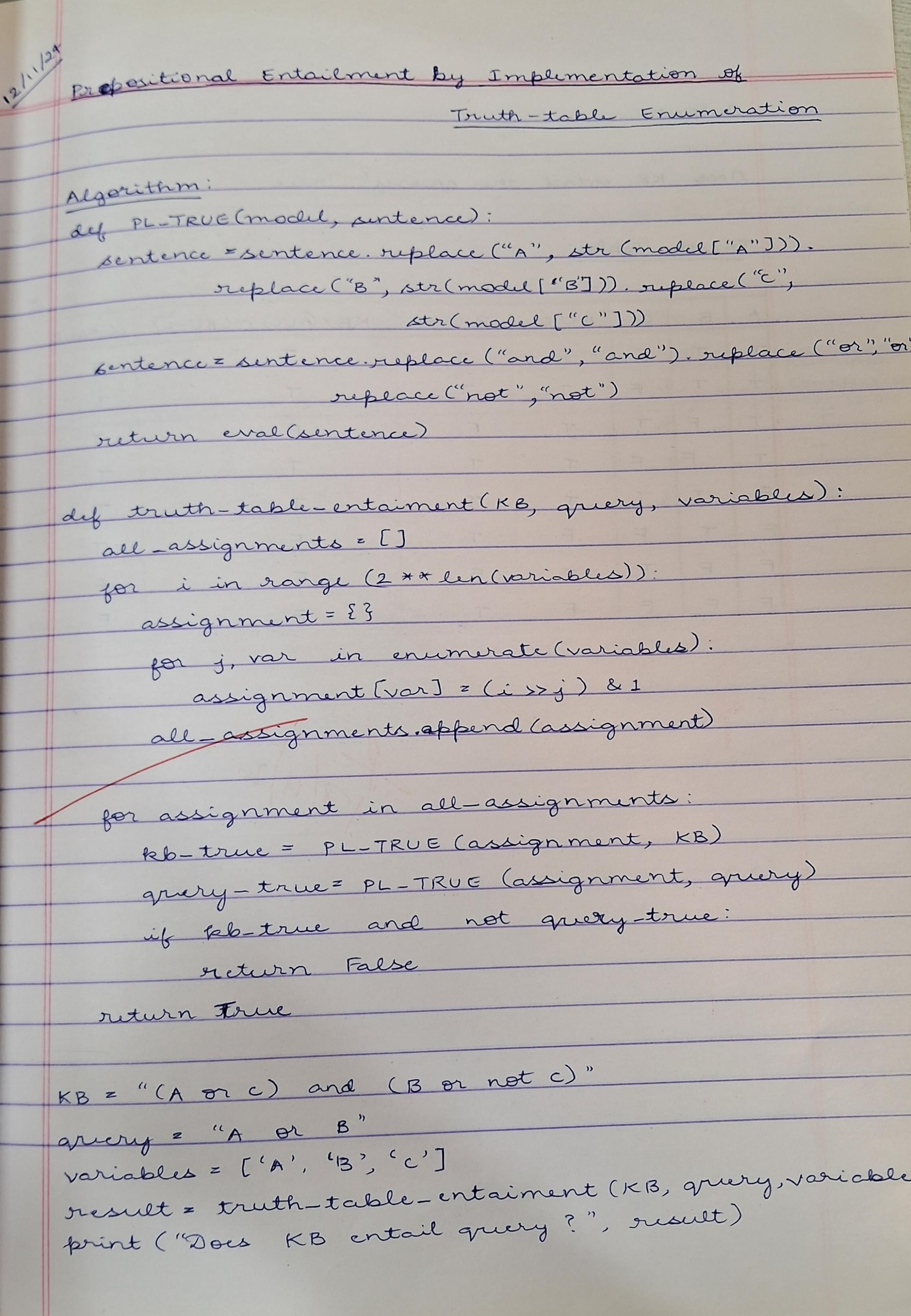
query = "A or B"

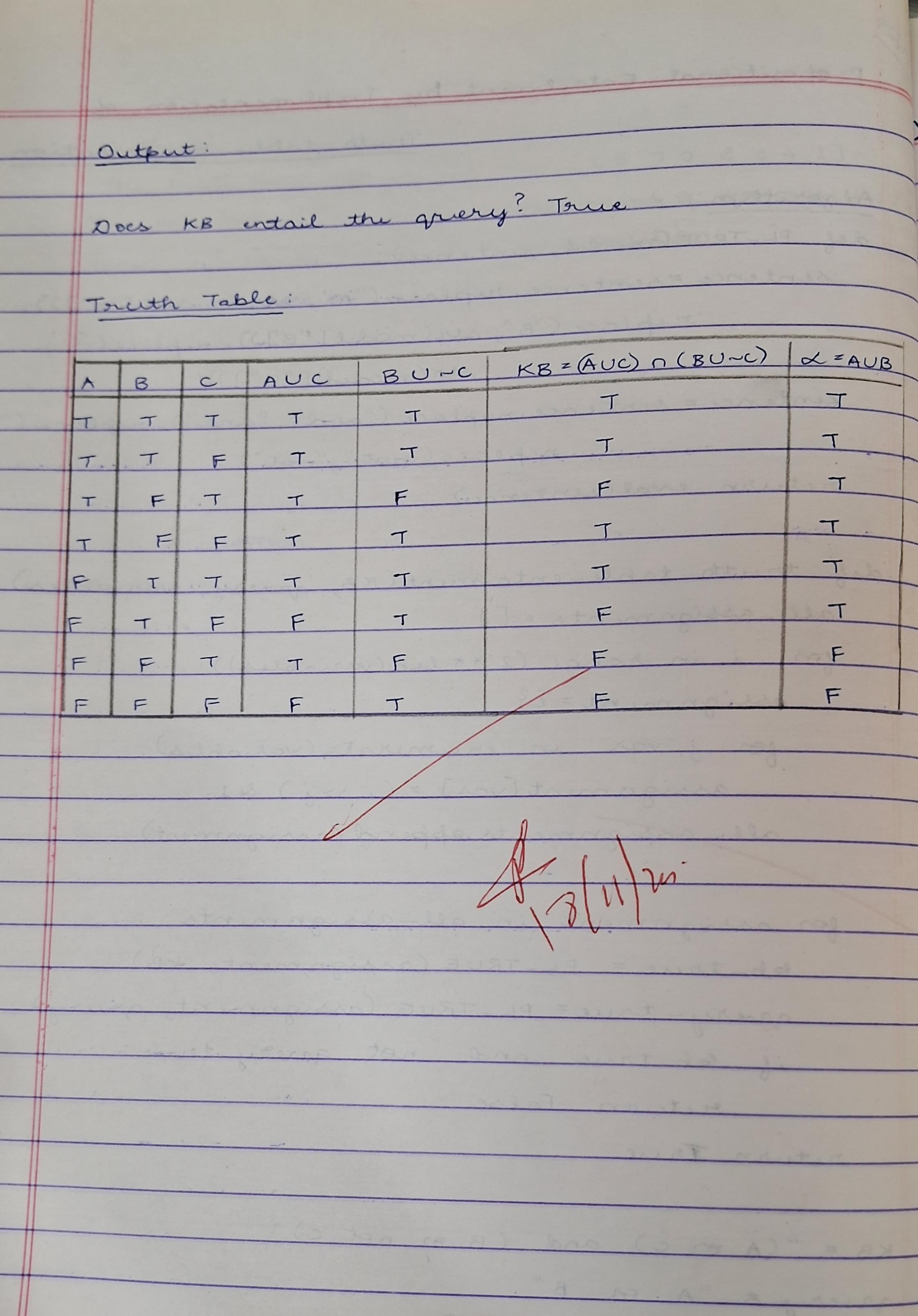
variables = ['A', 'B', 'C']

result = truth\_table\_entaiment(KB, query, variables)

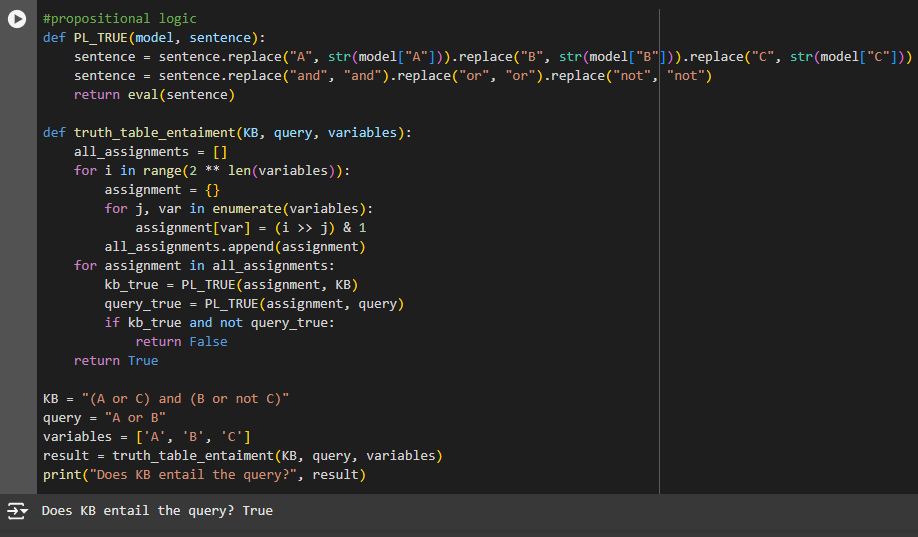
print("Does KB entail the query?", result)

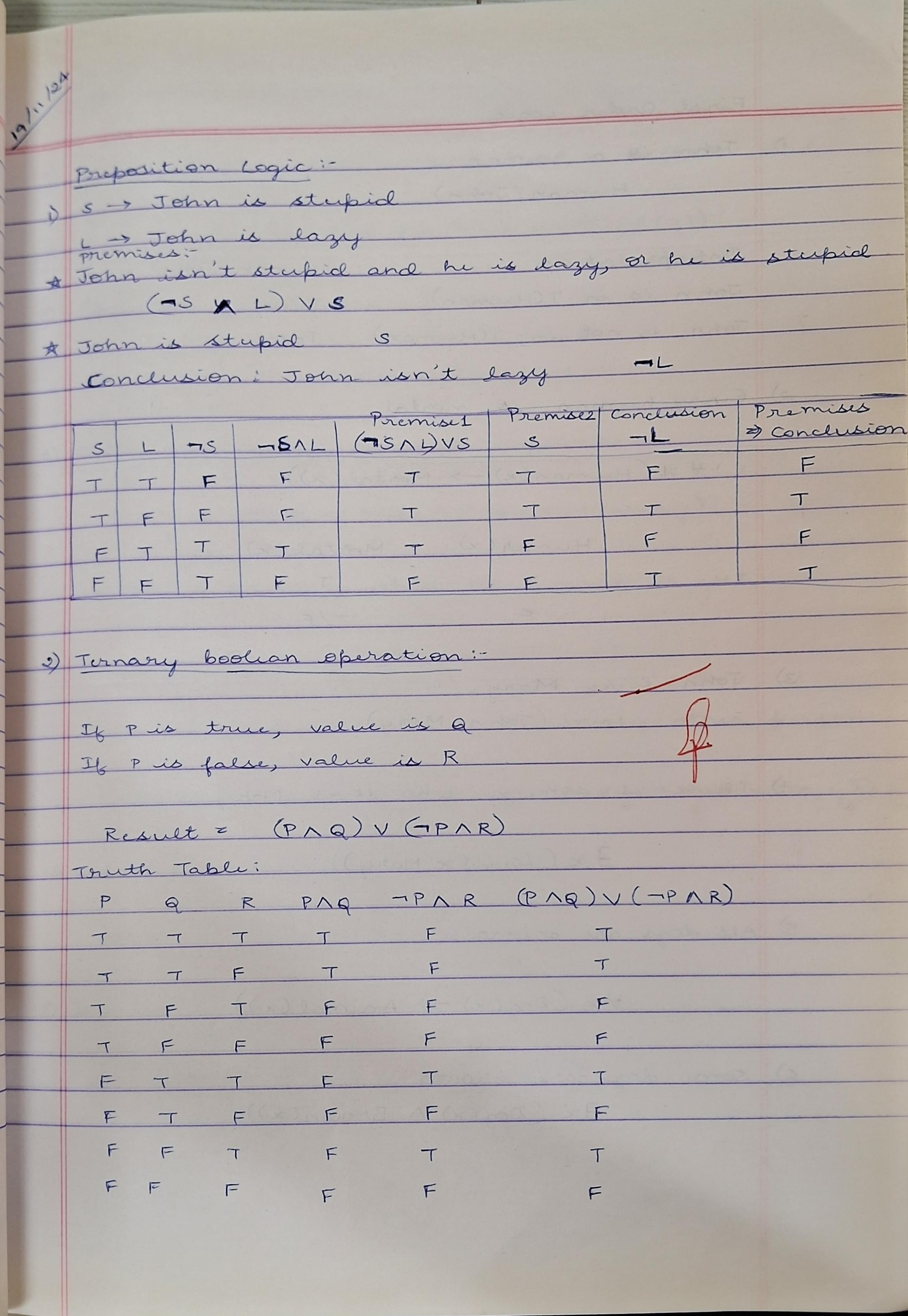
**Algorithm:**

****

****

**Output:**

****

****