**LAB 8:**

**Program:** Create knowledge base consisting of FOL statements and prove given query using forward reasoning.

**Code:**

class Fact:

    def \_\_init\_\_(self, predicate, args=None):

        self.predicate = predicate

        self.args = args if args else []

    def \_\_repr\_\_(self):

        return f"{self.predicate}({', '.join(map(str, self.args))})"

class Rule:

    def \_\_init\_\_(self, premise, conclusion):

        self.premise = premise

        self.conclusion = conclusion

    def \_\_repr\_\_(self):

        return f"IF {' AND '.join(map(str, self.premise))} THEN {self.conclusion}"

class KnowledgeBase:

    def \_\_init\_\_(self):

        self.facts = set()

        self.rules = []

    def add\_fact(self, fact):

        self.facts.add(fact)

    def add\_rule(self, rule):

        self.rules.append(rule)

    def forward\_reasoning(self, query):

        new\_facts = set(self.facts)

        derived = set()

        while True:

            added = False

            for rule in self.rules:

                if all(premise in new\_facts for premise in rule.premise):

                    if rule.conclusion not in new\_facts:

                        new\_facts.add(rule.conclusion)

                        derived.add(rule.conclusion)

                        added = True

            if not added:

                break

        return query in new\_facts

kb = KnowledgeBase()

kb.add\_fact(Fact("Food", ["Banana"]))

kb.add\_fact(Fact("Food", ["Pizza"]))

kb.add\_fact(Fact("Eats", ["Sam", "Idli"]))

kb.add\_fact(Fact("Alive", ["Sam"]))

kb.add\_fact(Fact("Enjoys", ["Ravi", "Food"]))

kb.add\_rule(Rule([Fact("Eats", ["X", "Y"]), Fact("Alive", ["X"])], Fact("Food", ["Y"])))

kb.add\_rule(Rule([Fact("Eats", ["Bill", "X"])], Fact("Eats", ["Sam", "X"])))

query = Fact("Enjoys", ["Ravi", "Idli"])

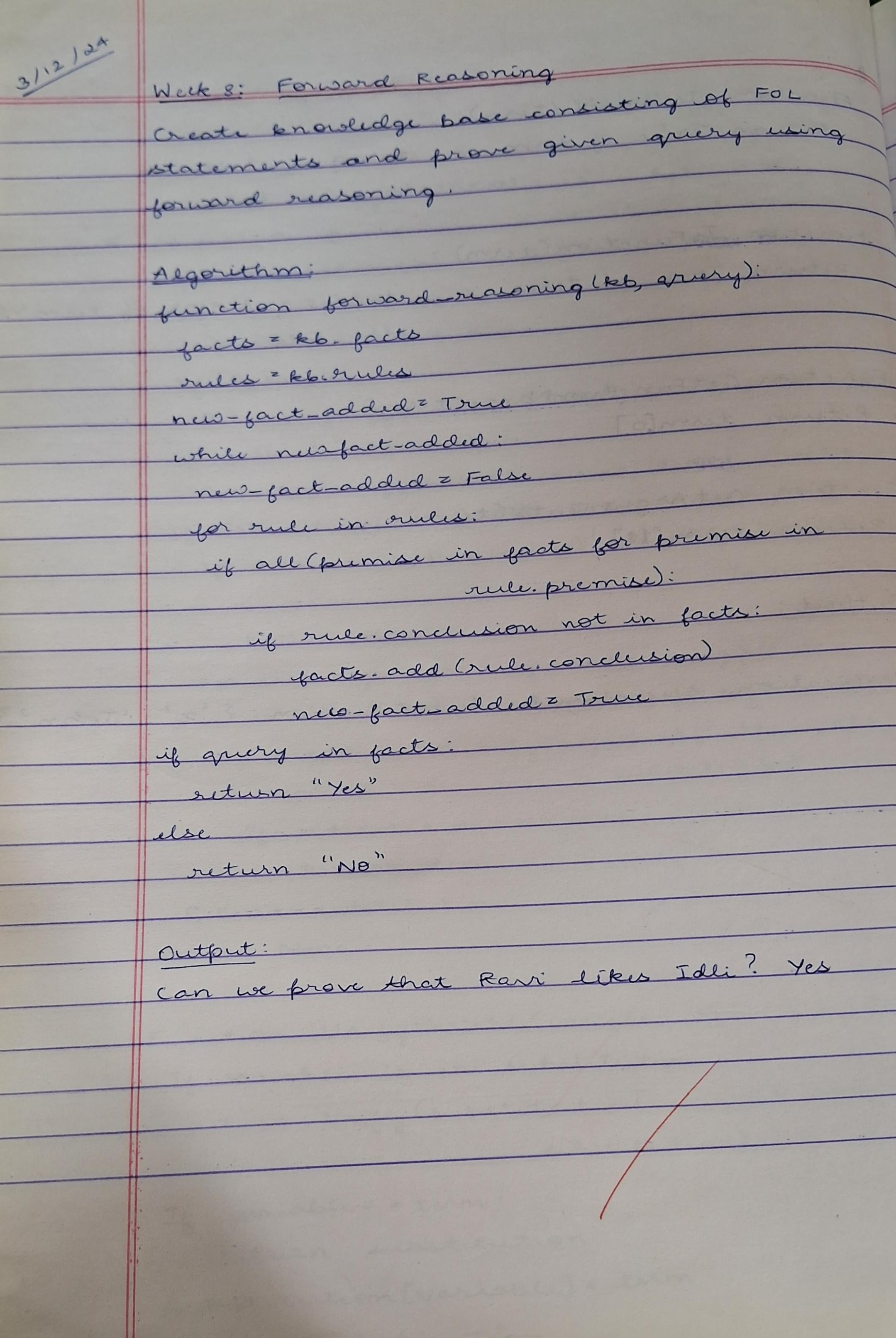
result = kb.forward\_reasoning(query)

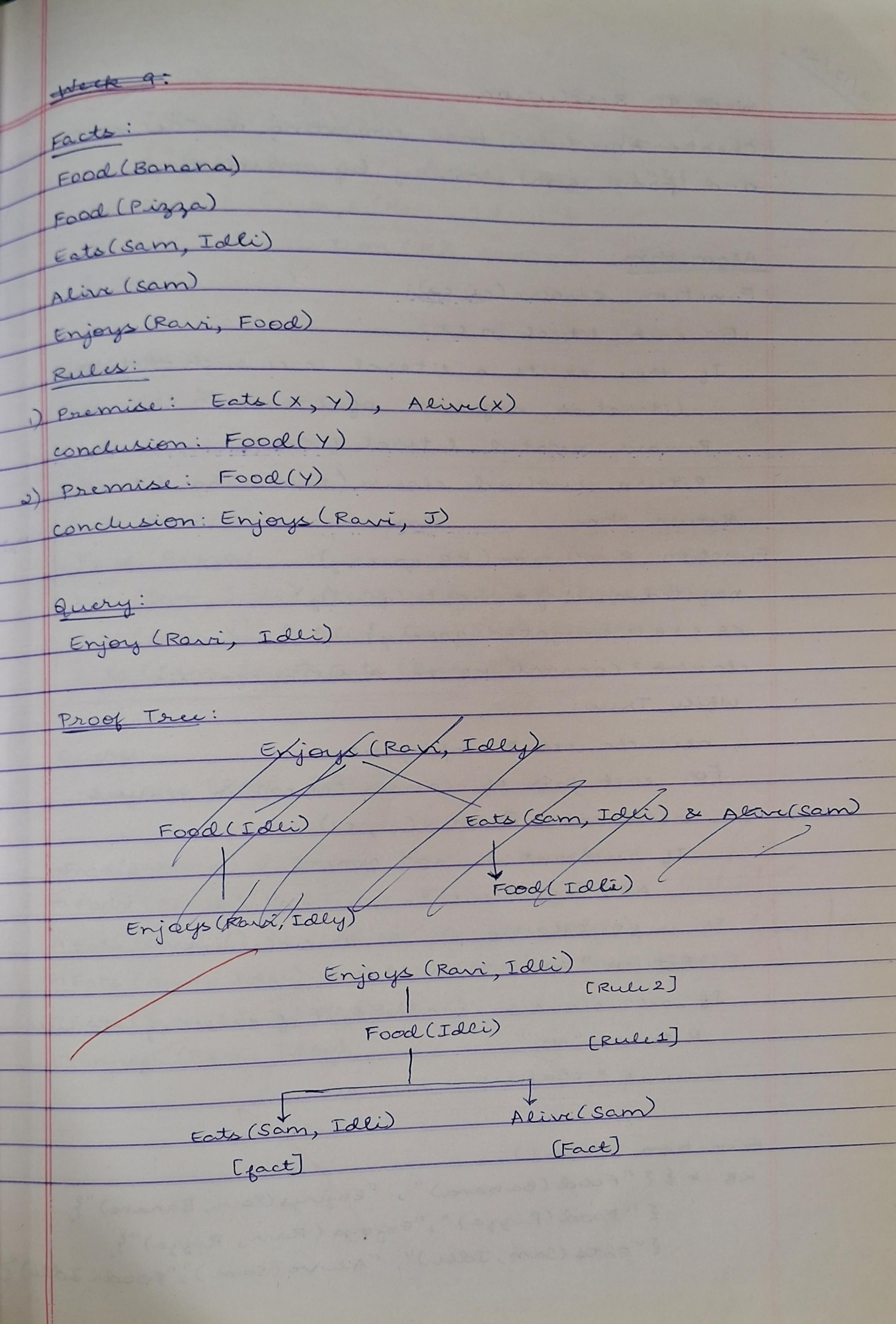
print(f"Can we prove that Ravi likes Idli? {'Yes' if result else 'No'}")

**Output:**

****

**Algorithm:**

****

****