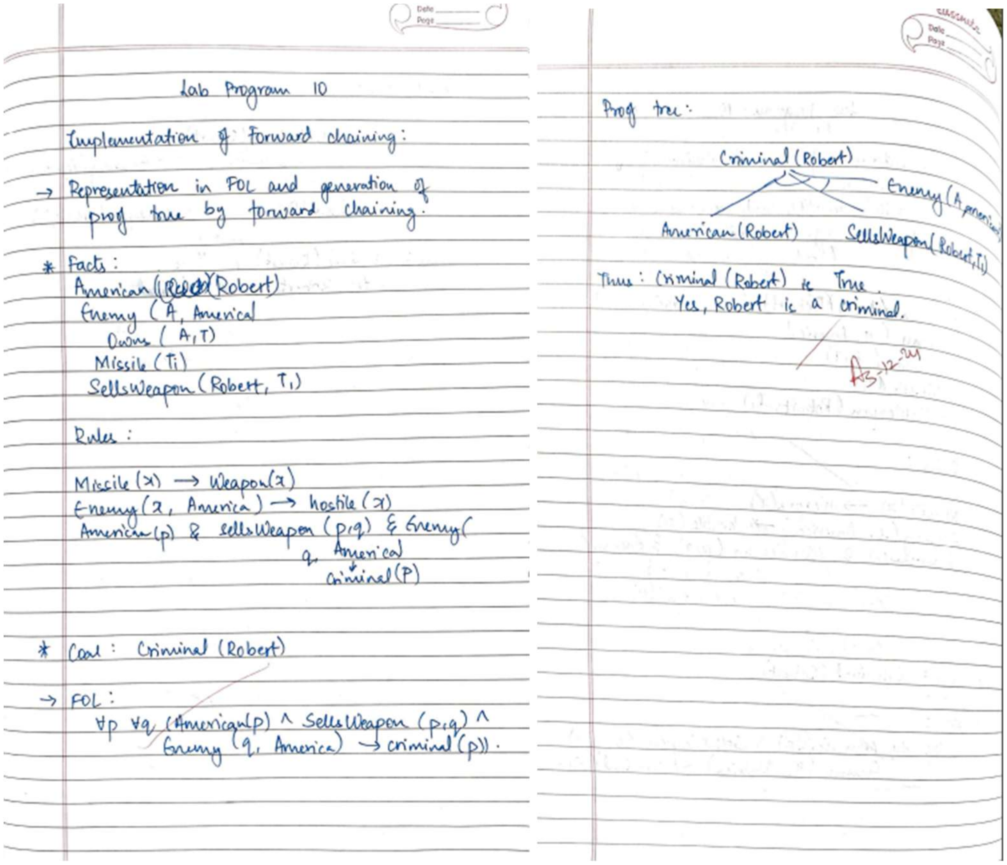
Program8:

Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning.

Algorithm:



Lab Program 10

Implementation of Forward chaining:

→ Representation in FOL and generation of prog tree by forward chaining.

* Facts:

American(~~(Robert~~(Robert)
Enemy (A, America)
Owns (A, T)
Missile (T1)
SellsWeapon (Robert, T1)

Rules:

Missile (x) → Weapon(x)
Enemy(x, America) → hostile (x)
American(p) & sellsWeapon (p,q) & Enemy(q, America)
↓
criminal(P)

* Goal: Criminal (Robert)

→ FOL:
∀p ∀q (American(p) ^ SellsWeapon (p,q) ^ Enemy (q, America) → criminal (p)).

Prog tree:

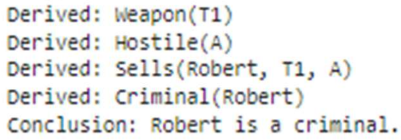
Thus: Criminal (Robert) is True.
Yes, Robert is a criminal.

A 3-12-24

Code:

```
class KnowledgeBaseSystem:
    def __init__(self):
        self.known_facts = set()
        self.rules_list = []

    def insert_fact(self, fact):
        self.known_facts.add(fact)

    def insert_rule(self, rule):
        self.rules_list.append(rule)

    def deduce(self):
        new_inferences = True
        while new_inferences:
            new_inferences = False
            for rule in self.rules_list:
                if rule.check_and_apply(self.known_facts):
                    new_inferences = True

knowledge_base = KnowledgeBaseSystem()
knowledge_base.insert_fact("American(Robert)")
knowledge_base.insert_fact("Missile(T1)")
knowledge_base.insert_fact("Owns(A, T1)")
knowledge_base.insert_fact("Enemy(A, America)")


class InferenceRule:
    def __init__(self, conditions, result):
        self.conditions = conditions  # List of conditions
        self.result = result  # The result to derive if conditions are met

    def check_and_apply(self, facts):
        if all(condition in facts for condition in self.conditions):
            if self.result not in facts:
                facts.add(self.result)
                print(f"Derived: {self.result}")
                return True
        return False



knowledge_base.insert_rule(InferenceRule(["Missile(T1)"],"Weapon(T1)"))
knowledge_base.insert_rule(InferenceRule(["Enemy(A, America)"],"Hostile(A)"))
knowledge_base.insert_rule(InferenceRule(["Missile(T1)", "Owns(A, T1)"],"Sells(Robert,
T1, A)"))
knowledge_base.insert_rule(InferenceRule(
    ["American(Robert)", "Weapon(T1)", "Sells(Robert, T1, A)",
"Hostile(A)"],"Criminal(Robert)"))
knowledge_base.deduce()

if "Criminal(Robert)" in knowledge_base.known_facts:
    print("Conclusion: Robert is a criminal.")
else:
    print("Conclusion: Unable to prove Robert is a criminal.")
```

Output Snapshot:



```
Derived: Weapon(T1)
Derived: Hostile(A)
Derived: Sells(Robert, T1, A)
Derived: Criminal(Robert)
Conclusion: Robert is a criminal.
```