Q). Create a knowledge base consisting of first order logic statements and

prove the given query using forward reasoning.

Code:

```
facts = {
    "American(Robert)": True,
    "Missile(T1)": True,
    "Enemy(A, America)": True,
    "Owns(A, T1)": True,
    "Hostile(A)": False,
    "Weapon(T1)": False,
    "Sells(Robert, T1, A)": False,
    "Criminal(Robert)": False,
}

rules = [
    ("American(Robert) and Weapon(T1) and Sells(Robert, T1, A) and Hostile(A)", "Criminal(Robert)"),
    ("Owns(A, T1) and Missile(T1)", "Weapon(T1)"),
    ("Missile(T1) and Owns(A, T1)", "Sells(Robert, T1, A)"),
    ("Enemy(A, America)", "Hostile(A)"),
]

def check_fact(fact):
    return facts.get(fact, False)

def parse_condition(condition):
    return condition.split(" and ")

def forward_reasoning():
    new_inferences = True
    while new_inferences:
        new_inferences = False
        for condition, conclusion in rules:
            condition_facts = parse_condition(condition)
            if all(check_fact(fact) for fact in condition_facts):
                if not check_fact(conclusion):
                    facts[conclusion] = True
                    new_inferences = True
                    print(f"Inferred: {conclusion}")

def print_inferred_facts():
    forward_reasoning()
    print("\nFinal Inferred Facts:")
    for fact, value in facts.items():
        print(f"{fact} is {'TRUE' if value else 'FALSE'}")

print_inferred_facts()
```

Output:

```
Inferred: Weapon(T1)
Inferred: Sells(Robert, T1, A)
Inferred: Hostile(A)
Inferred: Criminal(Robert)

Final Inferred Facts:
American(Robert) is TRUE
Missile(T1) is TRUE
Enemy(A, America) is TRUE
Owns(A, T1) is TRUE
Hostile(A) is TRUE
Weapon(T1) is TRUE
Sells(Robert, T1, A) is TRUE
Criminal(Robert) is TRUE
```