

**Program 7: Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning.**

```
# Define the knowledge base (KB) as a set of facts
```

```
KB = set()
```

```
# Premises based on the provided FOL problem
```

```
KB.add('American(Robert)')
```

```
KB.add('Enemy(America, A)')
```

```
KB.add('Missile(T1)')
```

```
KB.add('Owns(A, T1)')
```

```
# Define inference rules
```

```
def modus_ponens(fact1, fact2, conclusion):
```

```
    """ Apply modus ponens inference rule: if fact1 and fact2 are true, then conclude conclusion
    """
```

```
    if fact1 in KB and fact2 in KB:
```

```
        KB.add(conclusion)
```

```
        print(f"Inferred: {conclusion}")
```

```
def forward_chaining():
```

```
    """ Perform forward chaining to infer new facts until no more inferences can be made """
```

```
    # 1. Apply: Missile(x) → Weapon(x)
```

```
    if 'Missile(T1)' in KB:
```

```
        KB.add('Weapon(T1)')
```

```
        print(f"Inferred: Weapon(T1)")
```

```

# 2. Apply: Sells(Robert, T1, A) from Owns(A, T1) and Weapon(T1)

if 'Owns(A, T1)' in KB and 'Weapon(T1)' in KB:

    KB.add('Sells(Robert, T1, A)')

    print(f"Inferred: Sells(Robert, T1, A)")


# 3. Apply: Hostile(A) from Enemy(A, America)

if 'Enemy(America, A)' in KB:

    KB.add('Hostile(A)')

    print(f"Inferred: Hostile(A)")


# 4. Now, check if the goal is reached (i.e., if 'Criminal(Robert)' can be inferred)

if 'American(Robert)' in KB and 'Weapon(T1)' in KB and 'Sells(Robert, T1, A)' in KB and
'Hostile(A)' in KB:

    KB.add('Criminal(Robert)')

    print("Inferred: Criminal(Robert)")


# Check if we've reached our goal

if 'Criminal(Robert)' in KB:

    print("Robert is a criminal!")

else:

    print("No more inferences can be made.")


# Run forward chaining to attempt to derive the conclusion

forward_chaining()

```

```
Inferred: Weapon(T1)
Inferred: Sells(Robert, T1, A)
Inferred: Hostile(A)
Inferred: Criminal(Robert)
Robert is a criminal!
```