

Practical 2

```
# Define set A
A = {3, 6, 7}

# Define relation R (as a set of ordered pairs)
R = {(3, 3), (6, 6), (7, 7), (3, 6), (6, 7)}

# Check Reflexive: (a, a) must be in R for all a in A
def is_reflexive(A, R):
    for a in A:
        if (a, a) not in R:
            return False
    return True

# Check Antisymmetric: if (a, b) and (b, a) are in R, then a must be \
    equal to b
def is_antisymmetric(R):
    for (a, b) in R:
        if (b, a) in R and a != b:
            return False
    return True

# Check Transitive: if (a, b) in R and (b, c) in R, then (a, c) must \
    also be in R
def is_transitive(R):
    for (a, b1) in R:
        for (b2, c) in R:
            if b1 == b2 and (a, c) not in R:
                return False
    return True

# Check Partial Order: A relation is a partial order if it is \
    reflexive, antisymmetric, and transitive
def is_partial_order(A, R):
    return is_reflexive(A, R) and is_antisymmetric(R) and \
        is_transitive(R)
```

```
# Print results
print("Reflexive:", is_reflexive(A, R))
print("Antisymmetric:", is_antisymmetric(R))
print("Transitive:", is_transitive(R))
print("Partial Order:", is_partial_order(A, R))
Reflexive: True
Antisymmetric: True
Transitive: False
Partial Order: False
```