19) Write a python program for Diffie-Hellman protocol, each participant selects a secret number x and sends the other participant ax mod q for some public number a. What would happen if the participants sent each other xa for some public number a instead? Give at least one method Alice and Bob could use to agree on a key. Can Eve break your system without finding the secret numbers? Can Eve find the secret numbers?

PROGRAM:-

```
def modinv(a, m):
    """Modular inverse using Extended Euclidean Algorithm"""
    def egcd(a, b):
        if a == 0:
            return b, 0, 1
        g, y, x = egcd(b % a, a)
        return g, x - (b // a) * y, y

    g, x, _ = egcd(a, m)
    if g != 1:
        raise Exception('No modular inverse')
    return x % m

# Public values
q = 7919  # a large prime
a = 2     # primitive root modulo q

# Alice and Bob choose secrets
alice_secret = 1234
bob_secret = 5678

# ✅ Secure Diffie-Hellman exchange
A = pow(a, alice_secret, q)
B = pow(a, bob_secret, q)
```

```
# Shared keys
alice_key = pow(B, alice_secret, q)
bob_key = pow(A, bob_secret, q)

print("Secure Shared Key:", alice_key, "==", bob_key)

# ❌ Insecure variant: sending x * a mod q
A_insecure = (alice_secret * a) % q
B_insecure = (bob_secret * a) % q

# Eve can recover the secrets easily
a_inv = modinv(a, q)
alice_recovered = (A_insecure * a_inv) % q
bob_recovered = (B_insecure * a_inv) % q

print("\nInsecure exchange:")
print("Alice sends:", A_insecure)
print("Bob sends:", B_insecure)
print("Eve recovers Alice's secret:", alice_recovered)
print("Eve recovers Bob's secret:", bob_recovered)
```

OUTPUT:-

```
Secure Shared Key: 3697 == 3697

Insecure exchange:
Alice sends: 2468
Bob sends: 3437
Eve recovers Alice's secret: 1234
Eve recovers Bob's secret: 5678
```