```
def diffie_hellman():
    print("=== Diffie-Hellman Key Exchange ===")

    p = int(input("Enter a large prime number (p): "))
    g = int(input("Enter a primitive root modulo p (g): "))

    a = int(input("User A, enter your private key (a): "))
    b = int(input("User B, enter your private key (b): "))

    A = pow(g, a, p)  # A = g^a mod p
    B = pow(g, b, p)  # B = g^b mod p

    print(f"User A sends public key: {A}")
    print(f"User B sends public key: {B}")

    shared_key_a = pow(B, a, p)  # (B^a) mod p
    shared_key_b = pow(A, b, p)  # (A^b) mod p

    print(f"User A computes shared key: {shared_key_a}")
    print(f"User B computes shared key: {shared_key_b}")

    if shared_key_a == shared_key_b:
        print(f"\nShared secret established successfully! Key: {shared_key_a}")
    else:
        print("\nError: Keys do not match.")

diffie_hellman()
```

```
=== Diffie-Hellman Key Exchange ===
Enter a large prime number (p): 7
```