```
public class DiffieHellman {
  // Power function to return value of a ^ b mod P
  private static long calculatePower(long base, long exponent, long modulus) {
    if (exponent == 1)
      return base;
    else
      return (((long) Math.pow(base, exponent)) % modulus);
  }

  // Driver code
  public static void main(String[] args) {
    long prime, generator, x, privateKeyAlice, y, privateKeyBob, secretKeyAlice,
secretKeyBob;

    // Both parties agree upon the public keys generator and prime

    // A prime number prime is chosen
    prime = 23;
    System.out.println("Prime number (P): " + prime);

    // A primitive root for prime, generator is chosen
    generator = 5;
    System.out.println("Generator value (G): " + generator);

    // Alice chooses her private key privateKeyAlice
    privateKeyAlice = 6;
    System.out.println("Alice's private key (a): " + privateKeyAlice);

    // Gets the generated key
    x = calculatePower(generator, privateKeyAlice, prime);

    // Bob chooses his private key privateKeyBob
    privateKeyBob = 15;
    System.out.println("Bob's private key (b): " + privateKeyBob);

    // Gets the generated key
    y = calculatePower(generator, privateKeyBob, prime);

    // Generating the secret key after the exchange of keys
    secretKeyAlice = calculatePower(y, privateKeyAlice, prime); // Secret key for Alice
    secretKeyBob = calculatePower(x, privateKeyBob, prime); // Secret key for Bob

    System.out.println("Secret key for Alice: " + secretKeyAlice);
    System.out.println("Secret key for Bob: " + secretKeyBob);
  }
```

}

Output:
Prime number (P): 23
Generator value (G): 5
Alice's private key (a): 6
Bob's private key (b): 15
Secret key for Alice: 2
Secret key for Bob: 2