EX NO: **DIFFIE-HELLMAN KEY EXCHANGE** REG NO:210701290

DATE:

AIM:-

To implement a Diffie-Hellman key exchange algorithm using Java.

ALGORITHM:-

STEP 1:Both parties agree on and publicly share a prime number(p) and a base (g).

STEP 2:Each party generates a private key (a for Party A and b for Party B) randomly within a specified range.

STEP 3:Each party calculates its public key by raising the base (g) to the power of its private key modulo the prime number (p).

STEP 4:Both parties exchange their public keys.

STEP 5:Each party calculates the shared secret by raising the received public key to the power of its own private key modulo the prime number (p).

STEP 6:Both parties now have the same shared secret, which can be used as a symmetric encryption key.

STEP 7:The shared secret key can be used for secure communication between the two parties.

PROGRAM:-

```
import java.math.BigInteger;

import java.util.Random;

public class Main {

   public static void main(String[] args) {

        BigInteger p = BigInteger.probablePrime(7, new Random());
```

```
        BigInteger g = BigInteger.valueOf(5); // Example generator
        BigInteger a = new BigInteger(11, new Random());
        BigInteger b = new BigInteger(11, new Random());
        BigInteger A = g.modPow(a, p);
        System.out.println("Alice's public key: " + A);
        BigInteger B = g.modPow(b, p);
        System.out.println("Bob's public key: " + B);
        BigInteger K_a = B.modPow(a, p);
        BigInteger K_b = A.modPow(b, p);
        if (K_a.equals(K_b)) {
            System.out.println("Shared secret key: " + K_a);
            System.out.println("Key exchange successful!");
        } else {
            System.out.println("Error: Key exchange failed!");
        }}}
```

OUTPUT:-

```
Alice's public key: 23
Bob's public key: 39
Shared secret key: 48
Key exchange successful!
```

RESULT:-