Ex. No : 6 **Date :**	**Diffie-Hellman key exchange algorithm**

AIM:

To implement the Diffie-Hellman Key Exchange algorithm for a given problem .

ALGORITHM:

1. Alice and Bob publicly agree to use a modulus p = 23 and base g = 5 (which is a primitive root modulo 23).
2. Alice chooses a secret integer $\boldsymbol{a}$ = 4, then sends Bob $A = g^{a} \bmod p$
   - ○ $A = 5^{4} \bmod 23 = 4$
3. Bob chooses a secret integer $\boldsymbol{b}$ = 3, then sends Alice $B = g^{b} \bmod p$
   - ○ $B = 5^{3} \bmod 23 = 10$
4. Alice computes $\boldsymbol{s} = B^{a} \bmod p$
   - ○ $\boldsymbol{s} = 10^{4} \bmod 23 = 18$

5. Bob computes $\boldsymbol{s} = A^{b} \bmod p$
   - ○ $\boldsymbol{s} = 4^{3} \bmod 23 = 18$

6. Alice and Bob now share a secret (the number 18).

PROGRAM:

DiffieHellman.java

```
class DiffieHellman {
   public static void main(String args[]) {
      int p = 23; /* publicly known (prime number) */
      int g = 5; /* publicly known (primitive root) */
      int x = 4; /* only Alice knows this secret */
      int y = 3; /* only Bob knows this secret */
      double aliceSends = (Math.pow(g, x)) % p;
      double bobComputes = (Math.pow(aliceSends, y)) % p;
      double bobSends = (Math.pow(g, y)) % p;
      double aliceComputes = (Math.pow(bobSends, x)) % p;
      double sharedSecret = (Math.pow(g, (x * y))) % p;
      System.out.println("simulation of Diffie-Hellman key exchange algorithm\n--
-------------------------------------------");
      System.out.println("Alice Sends : " + aliceSends);
      System.out.println("Bob Computes : " + bobComputes);
      System.out.println("Bob Sends : " + bobSends);
```

```
        System.out.println("Alice Computes : " + aliceComputes);
        System.out.println("Shared Secret : " + sharedSecret);
        /* shared secrets should match and equality is transitive */
        if ((aliceComputes == sharedSecret) && (aliceComputes == bobComputes))
            System.out.println("Success: Shared Secrets Matches! " + sharedSecret);
        else
            System.out.println("Error: Shared Secrets does not Match");
    }
}
```

OUTPUT:

```
simulation of Diffie-Hellman key exchange algorithm
---------------------------------------------------------------
Alice Sends : 4.0
Bob Computes : 18.0
Bob Sends : 10.0
Alice Computes : 18.0
Shared Secret : 18.0
Success: Shared Secrets Matches! 18.0
```

RESULT:

Thus the *Diffie-Hellman key exchange algorithm* has been implemented using Java Program and the output has been verified successfully.