# Practical No.7

**Aim:-** To write a program to implement Diffie-Hellman key exchange technique for symmetric cryptography.

**Theory:-** Diffie-Hellman algorithm is being used to establish a shared secret that can be used for secret communications while exchanging data over a public network using the elliptic curve to generate points and get the secret key using the parameters ⇒

* For the sake of simplicity and practical implementation of algorithm consider only four variables, one prime P and G and two private values a and b.

* P and G are both publicly available numbers. Users pick private values a and b and they generate a key and exchange it publicly. The opposite person receives the key and that generates a ~~a~~ secret key, after which they have the same secret key to extract.

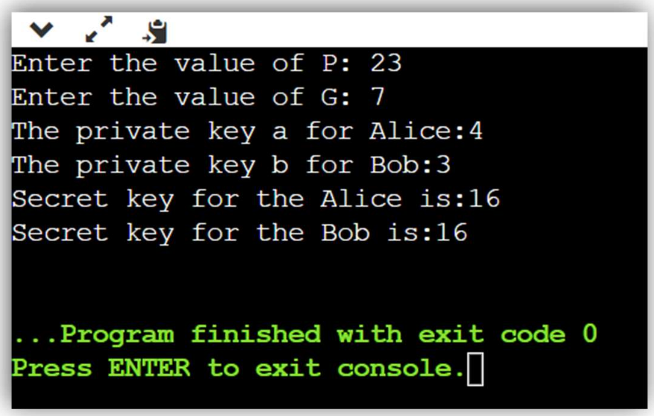
**Algorithm:-** The process step by step by step for user 1 (sender) and user 2 (receiver) are as follows:-

| | User 1 | User 2 |
|---|---|---|
| Step 1: | Public keys available<br>= P, G | Public key avaible<br>= P, G |
| Step 2: | Private key ~~s~~selected = a | Private key selected = b |
| Step 3: | Key generated:<br>$x = G^a \bmod P$ | Key generated:-<br>$y = G^b \bmod P$. |
| Step 4: | Exchange of generated keys ~~tt~~ takes place<br>key received = y. | Exchange of generated keys takes place<br>key ~~b~~ receives = x |

## Program:

```
import java.util.*;

class Main {
    // Power function to return value of a ^ b mod P
    private static long power(long a, long b, long p)
    {
        if (b == 1)
            return a;
        else
            return (((long)Math.pow(a, b)) % p);
    }

    // Driver code
    public static void main(String[] args)
    {
        long P, G, x, a, y, b, ka, kb;
         Scanner sc = new Scanner(System.in);
        // Both the persons will be agreed upon the
        // public keys G and P

        // A prime number P is taken
        System.out.print("Enter the value of P: ");
        P = sc.nextLong();

        // A primitive root for P, G is taken
        System.out.print("Enter the value of G: ");
        G = sc.nextLong();

        // Alice will choose the private key a
        // a is the chosen private key
        a = 4;
        System.out.println("The private key a for Alice:"+ a);

        // Gets the generated key
        x = power(G, a, P);

        // Bob will choose the private key b
        // b is the chosen private key
        b = 3;
        System.out.println("The private key b for Bob:"
                        + b);

        // Gets the generated key
        y = power(G, b, P);

        // Generating the secret key after the exchange
        // of keys
        ka = power(y, a, P); // Secret key for Alice
        kb = power(x, b, P); // Secret key for Bob

        System.out.println("Secret key for the Alice is:"+ ka);
        System.out.println("Secret key for the Bob is:"+ kb);
    }
}
```

**Output:**



```
Enter the value of P: 23
Enter the value of G: 7
The private key a for Alice:4
The private key b for Bob:3
Secret key for the Alice is:16
Secret key for the Bob is:16


...Program finished with exit code 0
Press ENTER to exit console.
```

**Conclusion:** The program to implement Diffie-Hellman key exchange technique for symmetric Cryptography has been executed successfully.