Security Lab
Lab Assignment No. 6

Aim: Design and Implement the Diffie Hellman Key exchange algorithm.

The **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 simplicity and practical implementation of the algorithm, we will consider only 4 variables, one prime P and G (a primitive root of P) and two private values a and b.
P and G are both publicly available numbers. Users (say Alice and Bob) 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 secret key, after which they have the same secret key to encrypt.

Algorithm:

STEP 1: Public keys of Alice and Bob is available = **P, G**
STEP 2: Private key of Alice is **a** and private key of Bob is **b**
STEP 3: Public key of Alice generated: **x** = G^a mod P
STEP 4: Public key of Bob generated: **y** = G^b mod P
STEP 5: Exchange of generated keys will take place
STEP 6: Key received to Alice will be **y** and key received to Bob will be **x**
STEP 7: Generated secret key of Alice will be **k_a** = y^a mod P
STEP 8: Generated secret key of Bob will be **k_b** = x^b mod P
STEP 9: Algebraically, it can be shown that k_a = k_b
STEP 10: Users will now have a symmetric secret key to encrypt

Code:

```
import random

# Both the persons will be agreed upon the public keys G and P
# A prime number P is taken
P = int(input("Enter a prime number for the public key: "))

# A primitive root for P, Q is taken
```

```
Q = int(input("Enter a base number for the public key: "))

print("\nThe value of G is: %d"%(P))
print("The value of N is: %d"%(Q))

# Alice will choose the private key a
a = int(input("\nEnter a private key for Alice: "))

# Bob will choose the private key b
b = int(input("Enter a private key for Bob: "))

# Generating a public key for Alice
alicePublic = int(pow(P, a, Q))

# Generating a public key for Bob
bobPublic = int(pow(P, b, Q))

print("\nThe public key of Alice: %d"%(alicePublic))
print("The public key of Bob: %d"%(bobPublic))

# Generating a secret key for Alice
aliceSecretKey = int(pow(bobPublic, a, P))

# Generating a secret key for Bob
bobSecretKey = int(pow(alicePublic, b, Q))

print("\nThe secret key of ALICE is :  %d"%(aliceSecretKey))
print("The secret key of BOB is :  %d"%(bobSecretKey))
```

Output:

```
PS D:\III Year Engineering\CNS Lab Experiments> & "C:/Users/Ninad Rao/AppData/Local/Programs/Python/Python39/python.exe" "d:/III Year Engineering/CNS Lab Experiments/assignment6.py"
Enter a prime number for the public key: 23
Enter a base number for the public key: 9

The value of P is: 23
The value of G is: 9

Enter a private key for Alice: 4
Enter a private key for Bob: 3

The private Key a for Alice is: 4
The private Key b for Bob is: 3

Secret key for the Alice is: 9
Secret Key for the Bob is: 9
```

Conclusion: Thus we understand how to design and implement the Diffie Hellman Key exchange algorithm.