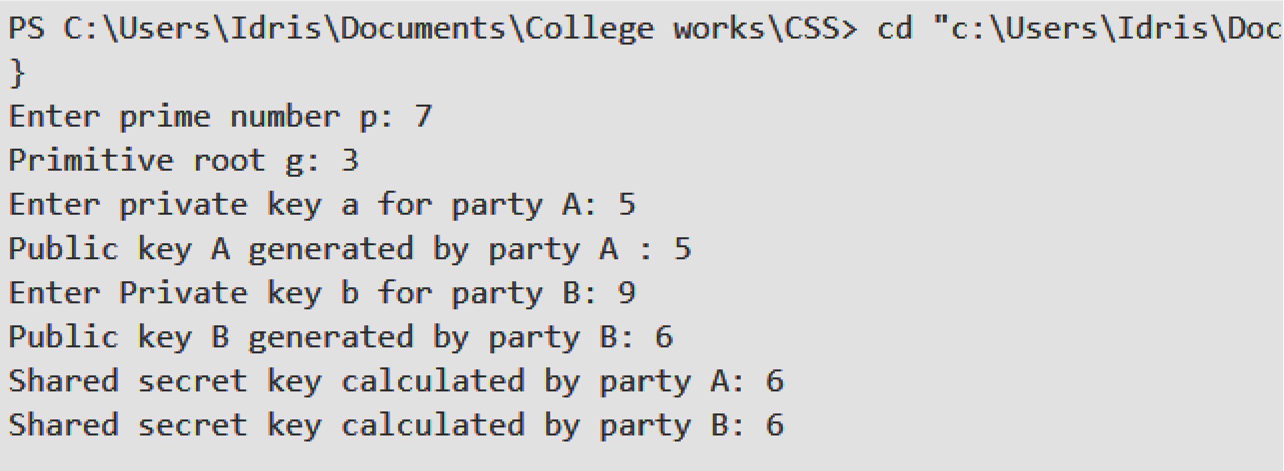
Program :

```
import java. math.Biglnteger; import java. util.Scanner;

public class DiffieHellman {
public static void main(String[l args) {
Scanner sc = new Scanner(System.in);

// Input prime number p
System.out.print("Enter prime number p: Il);
Biglnteger p = sc.nextBiglnteger();

// Find primitive root g
Biglnteger g = Biglnteger.valueOf(2); // start with 2
boolean found = false;
while (!found && g.compareTo(p) < O) {
Biglnteger x = Biglnteger.ONE;
for (Biglnteger i = Biglnteger.ZERO; i.compareTo(p.subtract(Biglnteger.TWO)) < O; i =i.add(Biglnteger.ONE)) {
x = x.multiply(g).mod(p);

if (x.equals(Biglnteger.ONE)) {
break;
} else if (i.equals(p.subtract(Biglnteger.valueOf(3)))) {
found = true;
break;
if (!found) { g = g.add(Biglnteger.ONE);

System.out.príntln("Primitive root g:”+ g);

// Input private key a for party A
System.out.print("Enter private key a for party A: “);
Biglnteger a = sc.nextBíglnteger();

// Calculate gna mod p
BigInteger A = g.modPow(a, p);

System.out.print("key A generated by party A:”);


// Input private key b for party B

System.out.print("Enter private key b for party B: ");

BigInteger b = sc.nextBígInteger();


// Calculate 0b mod p

BigInteger B = g.modPow(b, p);

System.out.println("Public key B generated by party B: " + B) ;


// Calculate shared secret key

BigInteger SA = B.modPow(a, p);

BigInteger SB = A.modPow(b, p);

System.out.println("Shared secret key calculated by party A :” +sA);

System.out.println("Shared secret key calculated by party B +sB);

sc.close();

}

}
```

Output :



```
PS C:\Users\Idris\Documents\College works\CSS> cd "c:\Users\Idris\Doc
}
Enter prime number p: 7
Primitive root g: 3
Enter private key a for party A: 5
Public key A generated by party A : 5
Enter Private key b for party B: 9
Public key B generated by party B: 6
Shared secret key calculated by party A: 6
Shared secret key calculated by party B: 6
```