**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.nextBígInteger();

// Calculate gna mod p

Biglnteger 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: ");

Biglnteger b = sc.nextBíglnteger();

// Calculate 0b mod p

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

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

// Calculate shared secret key

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

Biglnteger 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 :**

