

```

import java.math.BigInteger;
import java.util.Scanner;

public class RSA {

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

        System.out.print("Enter two distinct prime numbers (p and q): ");
        BigInteger p = sc.nextBigInteger();
        BigInteger q = sc.nextBigInteger();

        BigInteger n = p.multiply(q);
        BigInteger phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));

        BigInteger e = new BigInteger("3");
        while (!phi.gcd(e).equals(BigInteger.ONE)) {
            e = e.add(new BigInteger("2"));
        }

        BigInteger d = e.modInverse(phi);

        System.out.println("\nPublic key: (n=" + n + ", e=" + e + ")");
        System.out.println("Private key: (n=" + n + ", d=" + d + ")");

        System.out.print("\nEnter the message to encrypt (as a number): ");
        BigInteger message = sc.nextBigInteger();

        if (message.compareTo(n) >= 0) {
            System.out.println("Error: Message must be less than n.");
        } else {
            BigInteger ciphertext = message.modPow(e, n);
            System.out.println("Encrypted message: " + ciphertext);

            BigInteger decryptedMessage = ciphertext.modPow(d, n);
            System.out.println("Decrypted message: " + decryptedMessage);
        }

        sc.close();
    }
}

```