# Assignment No: 5 Roll No: 39

---

```
import java.util.Scanner;

public class RSA
{
// Function to compute (base^exp) % mod
static int modExp(int base, int exp, int mod) {
int result = 1;
base = base % mod;
while (exp > 0) {
if ((exp & 1) == 1) {
result = (result * base) % mod;
}
exp >>= 1;
base = (base * base) % mod;
}
return result;
}

// Function to compute GCD
static int gcd(int a, int b) {

while (b != 0) {
int temp = b;
b = a % b;
a = temp;
}
return a;
}

// Function to compute modular inverse
static int modInverse(int e, int phi) {
for (int d = 1; d < phi; d++) {
if ((e * d) % phi == 1) {
return d;
}
}
return -1; // No modular inverse found
}

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

```
// Input prime numbers
System.out.print("Enter first prime number (p): ");
int p = scanner.nextInt();
System.out.print("Enter second prime number (q): ");
int q = scanner.nextInt();
int n = p * q;
int phi = (p - 1) * (q - 1);

// Input e such that 1 < e < phi and gcd(e, phi) == 1
int e;
do {
System.out.print("Enter public exponent (e): ");
e = scanner.nextInt();
} while (e <= 1 || e >= phi || gcd(e, phi) != 1);
// Compute d (modular inverse of e mod phi)
int d = modInverse(e, phi);

System.out.println("Public Key: (" + e + ", " + n + ")");
System.out.println("Private Key: (" + d + ", " + n + ")");

// Input message
System.out.print("Enter message (as number): ");
int message = scanner.nextInt();
System.out.println("Original Message: " + message);

// Encryption: C = M^e mod n
int encrypted = modExp(message, e, n);
System.out.println("Encrypted: " + encrypted);

// Decryption: M = C^d mod n
int decrypted = modExp(encrypted, d, n);
System.out.println("Decrypted: " + decrypted);

scanner.close();
}
}
```

```
Run:
Enter first prime number (p): 7
Enter second prime number (q): 11
Enter public exponent (e): 7
Public Key: (7, 77)
Private Key: (43, 77)
Enter message (as number): 2
Original Message: 2
Encrypted: 51
Decrypted: 2
BUILD SUCCESSFUL (total time: 9 seconds)
```