Program

```
import java.util.*;

public class Rsa {


	static int gcd(int a, int b)
	{
		if (b == 0)
			return a;
		return gcd(b, a % b);
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		Scanner sc = new Scanner(System.in);
		int p,q;

		System.out.println("Enter two prime numbers");
		p = sc.nextInt();
		q = sc.nextInt();

		// First part of public key:
	    long n = p*q;

	    // Finding other part of public key e stands for encrypt
	    int e = 2;
	    long phi = (p-1)*(q-1);
	    while (e < phi){
	        // e must be co-prime to phi and smaller than phi.
	        if (gcd((int)e, (int)phi)==1)
	            break;
	        else
	            e++;
	    }
	    System.out.println(e);
	    // Private key (d stands for decrypt)
	    // choosing d such that it satisfies  d*e = 1 + k * totient
	    int k = 1;  // A constant value
	    double d1;
	    long d;
	    for(;;k++) {
			d1= (1 + (k*phi))/(double)e;
			if (d1 == Math.floor(d1))
			{
			   break;
			}
	    }

	    d = (long) d1;
```

```
        // Message to be encrypted
        System.out.println("Enter a small integer");
        int msg = sc.nextInt();
        System.out.println("Message data:" +  msg);

        // Encryption c = (msg ^ e) % n
        long c = (long) Math.pow(msg, e);
        c = c % n;
        System.out.println("Encrypted data:"+ c);

        //Decryption m = (c ^ d) % n
        long m = (long) (Math.pow(c, d));
        m = m % n;
        System.out.println("Original Message Sent:"+ m);

            sc.close();
    }

}
```

Output: