Implementation of RSA Algorithm

```
import java.math.*;
import java.util.*;

class RSA {
	public static void main(String args[])
	{
		int p, q, n, z, d = 0, e, i;

		// The number to be encrypted and decrypted
		int msg = 12;
		double c;
		BigInteger msgback;

		// 1st prime number p
		p = 3;

		// 2nd prime number q
		q = 11;
		n = p * q;
		z = (p - 1) * (q - 1);
		System.out.println("the value of z = " + z);

		for (e = 2; e < z; e++) {

			// e is for public key exponent
			if (gcd(e, z) == 1) {
				break;
			}
		}
		System.out.println("the value of e = " + e);
		for (i = 0; i <= 9; i++) {
			int x = 1 + (i * z);

			// d is for private key exponent
			if (x % e == 0) {
				d = x / e;
				break;
			}
		}
		System.out.println("the value of d = " + d);
		c = (Math.pow(msg, e)) % n;
		System.out.println("Encrypted message is : " + c);
```

```
                // converting int value of n to BigInteger
                BigInteger N = BigInteger.valueOf(n);

                // converting float value of c to BigInteger
                BigInteger C = BigDecimal.valueOf(c).toBigInteger();
                msgback = (C.pow(d)).mod(N);
                System.out.println("Decrypted message is : "
                                                + msgback);
        }

        static int gcd(int e, int z)
        {
                if (e == 0)
                        return z;
                else
                        return gcd(z % e, e);
        }
}
```