**import** java.util.Scanner;

**public** **class** RSAMain {

**public** **static** **void** main(String[] args) {

**int** p, q, e, msg;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print ("Enter any prime no. p = ");

p = sc.nextInt();

System.***out***.print ("Enter any prime no. q = ");

q = sc.nextInt();

System.***out***.print ("Enter e = ");

e = sc.nextInt();

System.***out***.print ("Enter plain text (msg) = ");

msg = sc.nextInt();

RSAcalculate calc = **new** RSAcalculate(p,q,e,msg);

calc.calculate();

}

}

**import** java.math.BigDecimal;

**import** java.math.BigInteger;

**public** **class** RSAcalculate {

**private** **int** p;

**private** **int** q;

**private** **int** e;

**private** **int** msg;

**public** RSAcalculate(**int** p, **int** q, **int** e, **int** msg) {

**this**.p = p;

**this**.q = q;

**this**.e = e;

**this**.msg = msg;

}

**public** **void** calculate()

{

**int** c;

BigInteger msgback;

**int** n, phi, d = 0, i;

n = p \* q;

phi = (p - 1) \* (q - 1);

System.***out***.println("the value of n = " + n);

System.***out***.println("the value of phi = " + phi);

**if**(e==0)

{

**for** (e = 2; e < phi; e++)

{

**if** (*gcd*(e, phi) == 1)

{

**break**;

}

}

}

System.***out***.println("the value of e = " + e);

**for** (i = 0; i <= e; i++)

{

**int** x = 1 + (i \* phi); //d = (1+(phi\*i))/ e

**if** (x % e == 0)

{

d = x / e;

**break**;

}

}

c = (**int**) ((Math.*pow*(msg, e)) % n); //c= p^e mod n

System.***out***.println("Encrypted message is : " + c);

System.***out***.println("the value of d = " + d);

// 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); //p= c^d mod n

System.***out***.println("Decrypted message is : "+ msgback);

}

**static** **int** gcd(**int** e, **int** phi)

{

**if** (e == 0)

**return** phi;

**else**

**return** *gcd*(phi % e, e);

}

}