# **Lab 20: Fast RSA Decryption**

**Group Name: Srikar Maadhu**

**Lahari Karrotu**

Implement FAST RSA Decryption with optimized CRT(1).

*If mp<mq, compute*

dP= d mod (p-1),

dQ= d mod (q-1),

qInv = q^-1 mod p

Let:

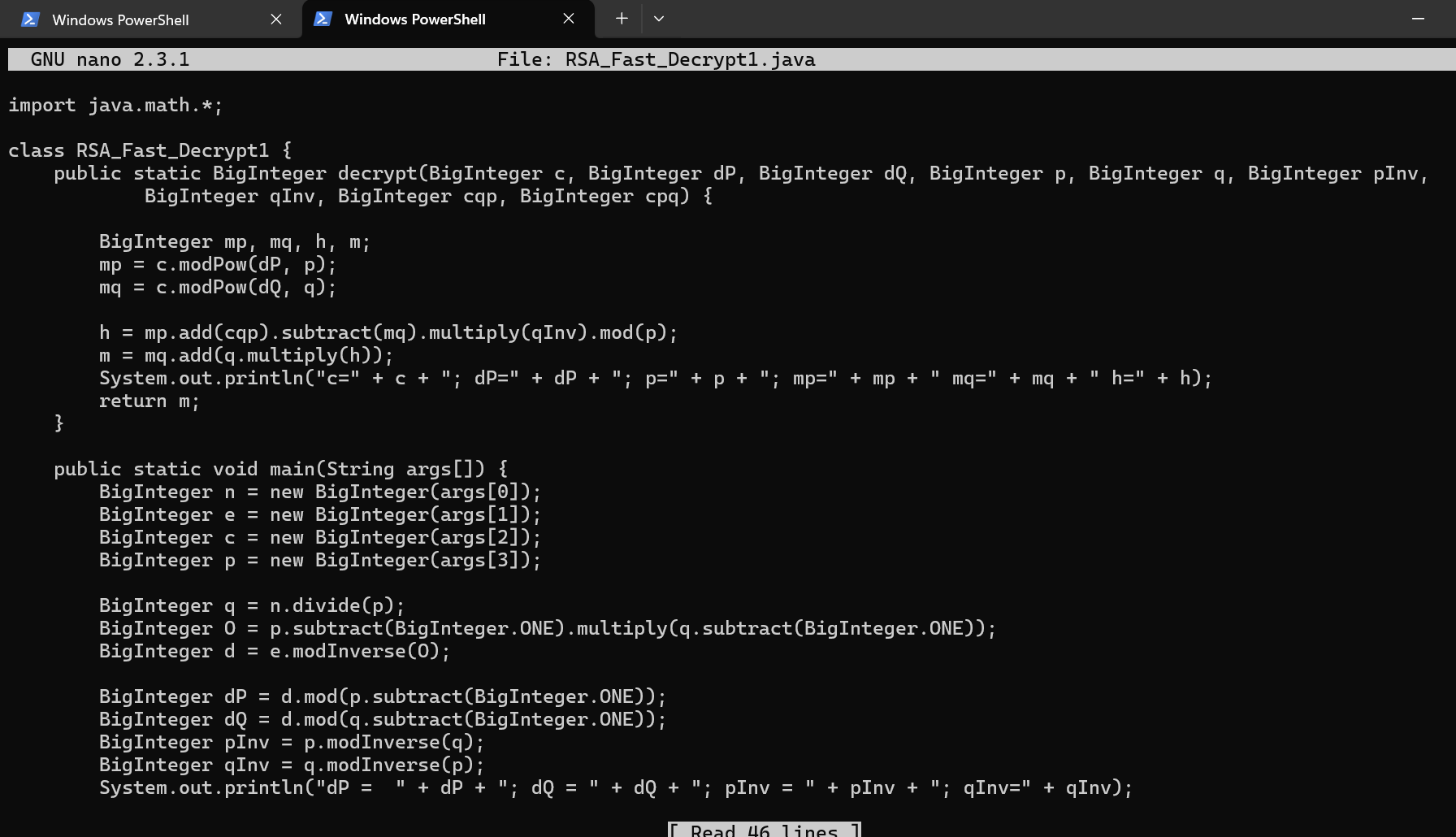
mp = c^dP mod p

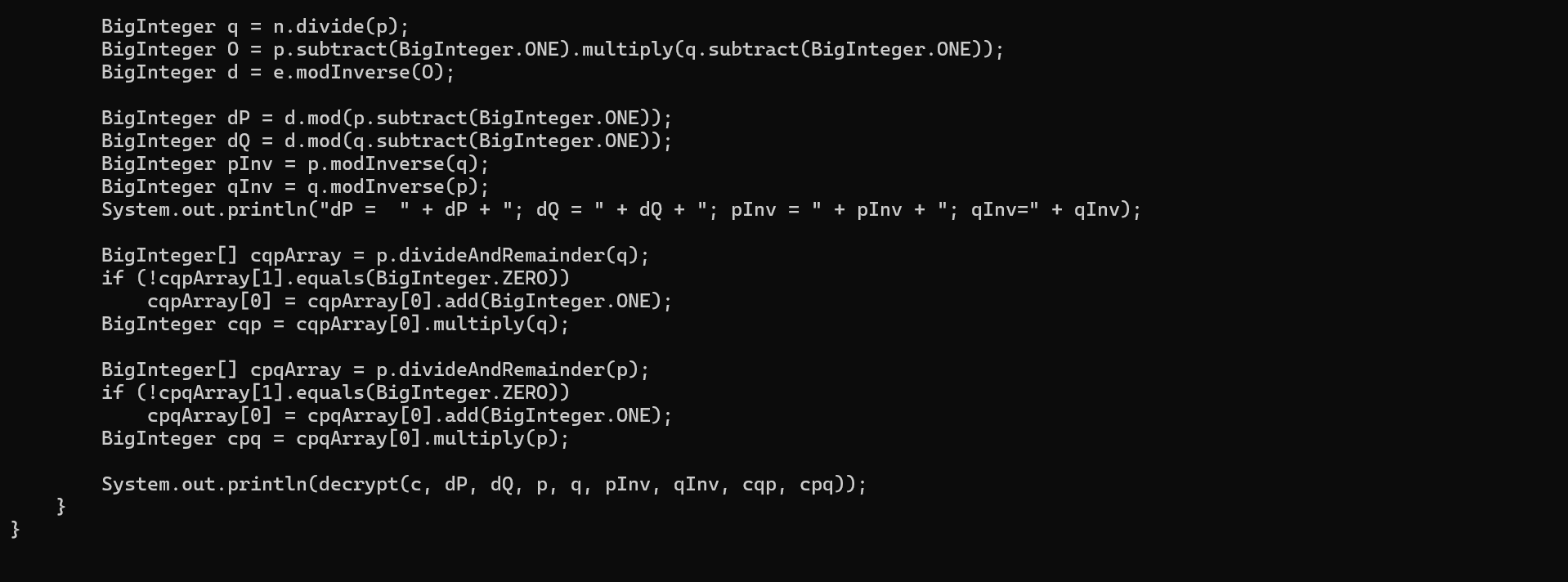
mq = c^dQ mod q.

h = (mp + ceil[q/p]p - mq) \* qInv (mod p)

Let

m = mq + q \* h

As template you may use the following, adding print statements to compare with manual computations at each step, for debugging and understanding the process: 



Code:

import java.math.\*;

class RSA\_Fast\_Decrypt1 {

public static BigInteger decrypt(BigInteger c, BigInteger dP, BigInteger dQ, BigInteger p, BigInteger q, BigInteger pInv,

BigInteger qInv, BigInteger cqp, BigInteger cpq) {

BigInteger mp, mq, h, m;

mp = c.modPow(dP, p);

mq = c.modPow(dQ, q);

h = mp.add(cqp).subtract(mq).multiply(qInv).mod(p);

m = mq.add(q.multiply(h));

System.out.println("c=" + c + "; dP=" + dP + "; p=" + p + "; mp=" + mp + " mq=" + mq + " h=" + h);

return m;

}

public static void main(String args[]) {

BigInteger n = new BigInteger(args[0]);

BigInteger e = new BigInteger(args[1]);

BigInteger c = new BigInteger(args[2]);

BigInteger p = new BigInteger(args[3]);

BigInteger q = n.divide(p);

BigInteger O = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));

BigInteger d = e.modInverse(O);

BigInteger dP = d.mod(p.subtract(BigInteger.ONE));

BigInteger dQ = d.mod(q.subtract(BigInteger.ONE));

BigInteger pInv = p.modInverse(q);

BigInteger qInv = q.modInverse(p);

System.out.println("dP = " + dP + "; dQ = " + dQ + "; pInv = " + pInv + "; qInv=" + qInv);

BigInteger[] cqpArray = p.divideAndRemainder(q);

if (!cqpArray[1].equals(BigInteger.ZERO))

cqpArray[0] = cqpArray[0].add(BigInteger.ONE);

BigInteger cqp = cqpArray[0].multiply(q);

BigInteger[] cpqArray = p.divideAndRemainder(p);

if (!cpqArray[1].equals(BigInteger.ZERO))

cpqArray[0] = cpqArray[0].add(BigInteger.ONE);

BigInteger cpq = cpqArray[0].multiply(p);

System.out.println(decrypt(c, dP, dQ, p, q, pInv, qInv, cqp, cpq));

}

}

Test on Dec(11,[7,11\*17])=88.

For testing, note that inv of d= 7 is e = 23.

Also for tests, note that pInv=14, qInv=2, mp=0, mq=3, h=5.

java RSA\_Fast\_Decrypt 187 23 11 11

88

