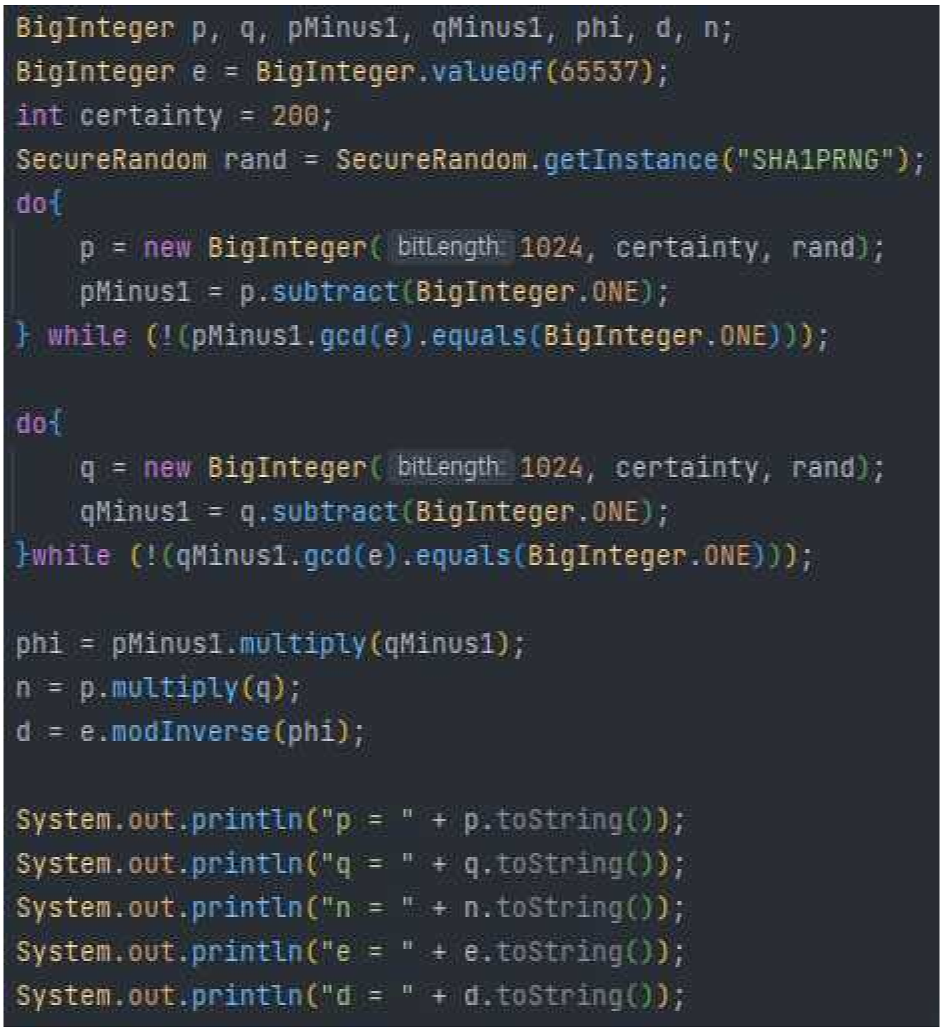
정보보호이론 [과제-2]

목차

[과제-2]A - (1)

RSA 공개키 암호 parameters(p, q, n, e, d) 생성하기

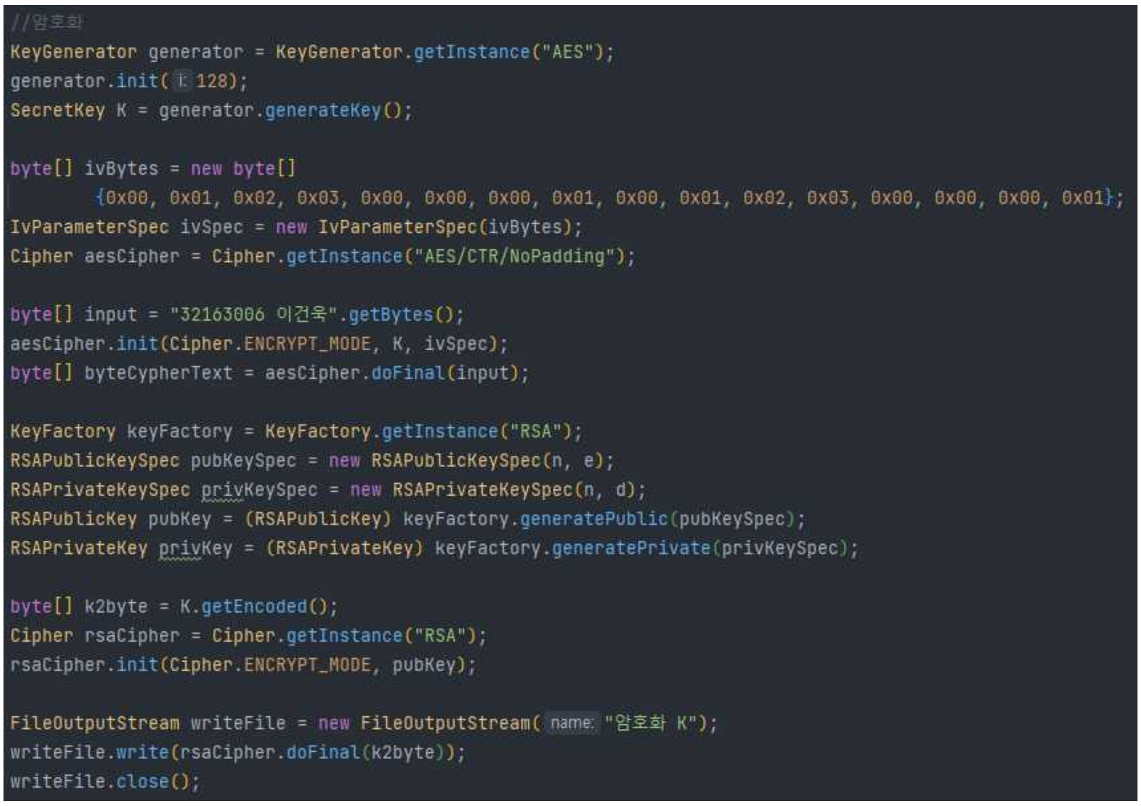


```java
BigInteger p, q, pMinus1, qMinus1, phi, d, n;
BigInteger e = BigInteger.valueOf(65537);
int certainty = 200;
SecureRandom rand = SecureRandom.getInstance("SHA1PRNG");
do{
    p = new BigInteger( bitLength: 1024, certainty, rand);
    pMinus1 = p.subtract(BigInteger.ONE);
} while (!(pMinus1.gcd(e).equals(BigInteger.ONE)));

do{
    q = new BigInteger( bitLength: 1024, certainty, rand);
    qMinus1 = q.subtract(BigInteger.ONE);
}while (!(qMinus1.gcd(e).equals(BigInteger.ONE)));

phi = pMinus1.multiply(qMinus1);
n = p.multiply(q);
d = e.modInverse(phi);

System.out.println("p = " + p.toString());
System.out.println("q = " + q.toString());
System.out.println("n = " + n.toString());
System.out.println("e = " + e.toString());
System.out.println("d = " + d.toString());
```

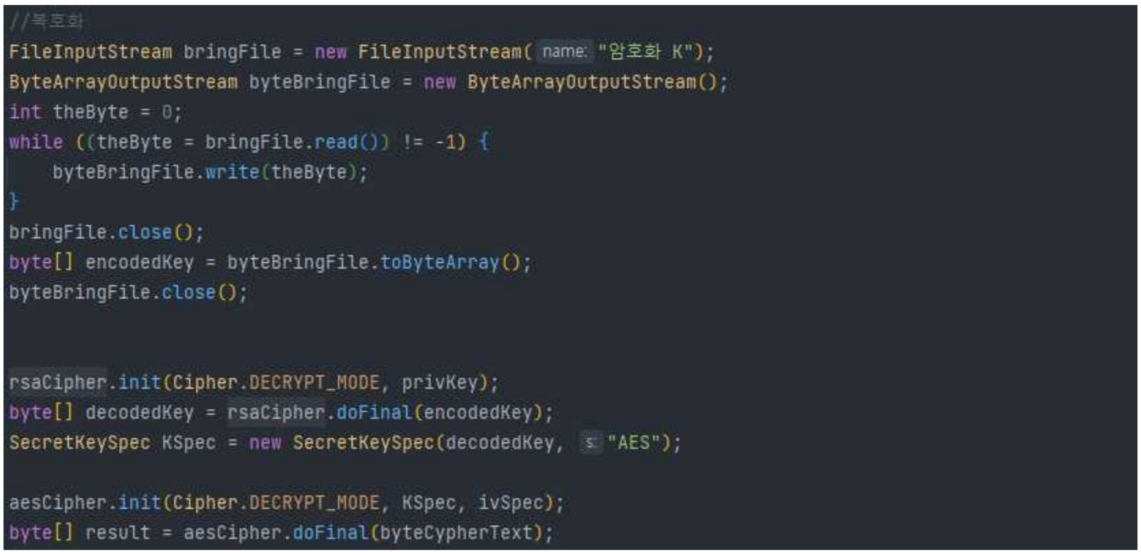
[과제-2]A – (2), (3)

대칭키 K 생성해서 RSA 암호화 하고, 파일에 저장 및 파일로부터 불러들여 복호화 하기.

평문 (본인의 성명 및 학번으로 구성)을 대칭키 암호화 및 복호화 하기.

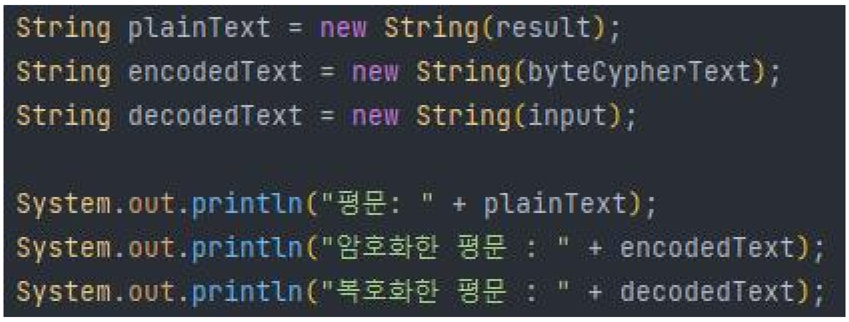


```java
//암호화
KeyGenerator generator = KeyGenerator.getInstance("AES");
generator.init( l: 128);
SecretKey K = generator.generateKey();

byte[] ivBytes = new byte[]
        {0x00, 0x01, 0x02, 0x03, 0x00, 0x00, 0x00, 0x01, 0x00, 0x01, 0x02, 0x03, 0x00, 0x00, 0x00, 0x01};
IvParameterSpec ivSpec = new IvParameterSpec(ivBytes);
Cipher aesCipher = Cipher.getInstance("AES/CTR/NoPadding");

byte[] input = "32163006 이건욱".getBytes();
aesCipher.init(Cipher.ENCRYPT_MODE, K, ivSpec);
byte[] byteCypherText = aesCipher.doFinal(input);

KeyFactory keyFactory = KeyFactory.getInstance("RSA");
RSAPublicKeySpec pubKeySpec = new RSAPublicKeySpec(n, e);
RSAPrivateKeySpec privKeySpec = new RSAPrivateKeySpec(n, d);
RSAPublicKey pubKey = (RSAPublicKey) keyFactory.generatePublic(pubKeySpec);
RSAPrivateKey privKey = (RSAPrivateKey) keyFactory.generatePrivate(privKeySpec);

byte[] k2byte = K.getEncoded();
Cipher rsaCipher = Cipher.getInstance("RSA");
rsaCipher.init(Cipher.ENCRYPT_MODE, pubKey);

FileOutputStream writeFile = new FileOutputStream( name: "암호화 K");
writeFile.write(rsaCipher.doFinal(k2byte));
writeFile.close();
```

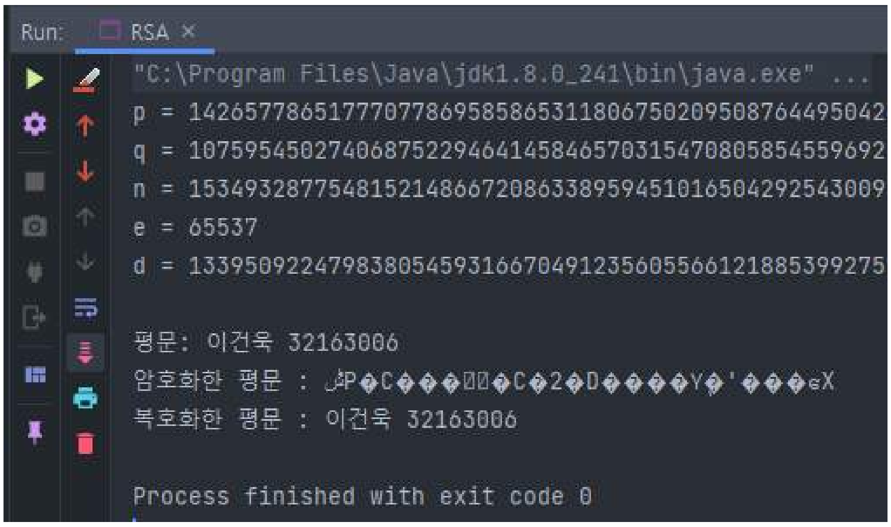


```java
//복호화
FileInputStream bringFile = new FileInputStream( name: "암호화 K");
ByteArrayOutputStream byteBringFile = new ByteArrayOutputStream();
int theByte = 0;
while ((theByte = bringFile.read()) != -1) {
    byteBringFile.write(theByte);
}
bringFile.close();
byte[] encodedKey = byteBringFile.toByteArray();
byteBringFile.close();


rsaCipher.init(Cipher.DECRYPT_MODE, privKey);
byte[] decodedKey = rsaCipher.doFinal(encodedKey);
SecretKeySpec KSpec = new SecretKeySpec(decodedKey,  s: "AES");

aesCipher.init(Cipher.DECRYPT_MODE, KSpec, ivSpec);
byte[] result = aesCipher.doFinal(byteCypherText);
```

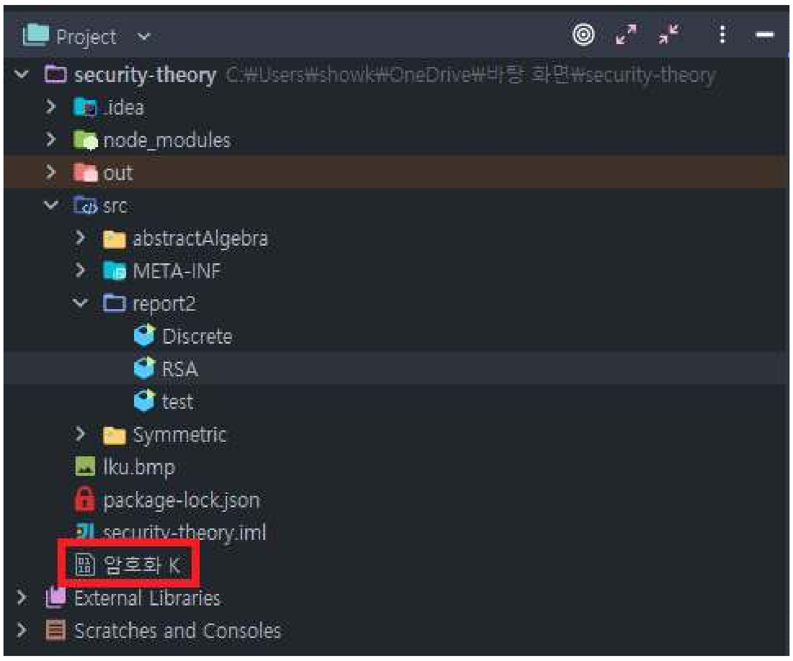
[과제-2]A – 실행결과



```
String plainText = new String(result);
String encodedText = new String(byteCypherText);
String decodedText = new String(input);

System.out.println("평문: " + plainText);
System.out.println("암호화한 평문 : " + encodedText);
System.out.println("복호화한 평문 : " + decodedText);
```

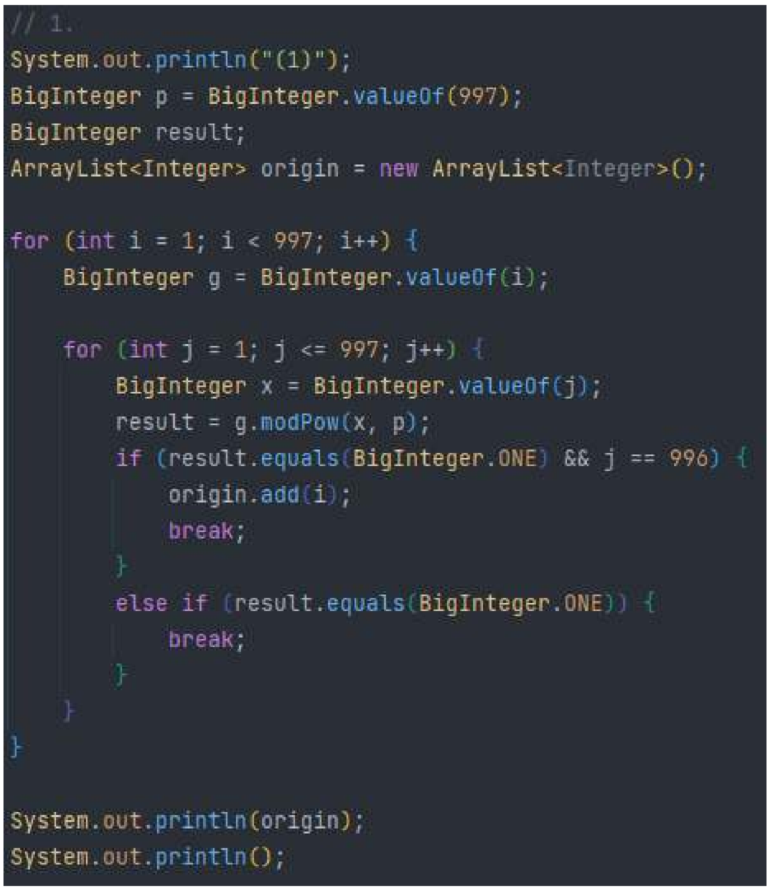


```
"C:\Program Files\Java\jdk1.8.0_241\bin\java.exe" ...
p = 14265778651777077869585865311806750209508764495042
q = 10759545027406875229464145846570315470805854559692
n = 15349328775481521486672086338959451016504292543009
e = 65537
d = 13395092247983805459316670491235605566121885399275

평문: 이건욱 32163006
암호화한 평문 : ꟼP�C���⃞⃞�C�2�D����Y�'���eX
복호화한 평문 : 이건욱 32163006

Process finished with exit code 0
```

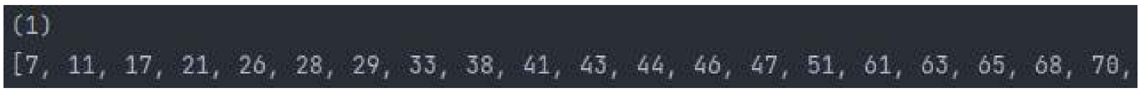


[과제-2]B – (1)

소수 p = 997에 대해서 “생성자” 모두 구하기



```java
// 1.
System.out.println("(1)");
BigInteger p = BigInteger.valueOf(997);
BigInteger result;
ArrayList<Integer> origin = new ArrayList<Integer>();

for (int i = 1; i < 997; i++) {
    BigInteger g = BigInteger.valueOf(i);

    for (int j = 1; j <= 997; j++) {
        BigInteger x = BigInteger.valueOf(j);
        result = g.modPow(x, p);
        if (result.equals(BigInteger.ONE) && j == 996) {
            origin.add(i);
            break;
        }
        else if (result.equals(BigInteger.ONE)) {
            break;
        }
    }
}

System.out.println(origin);
System.out.println();
```



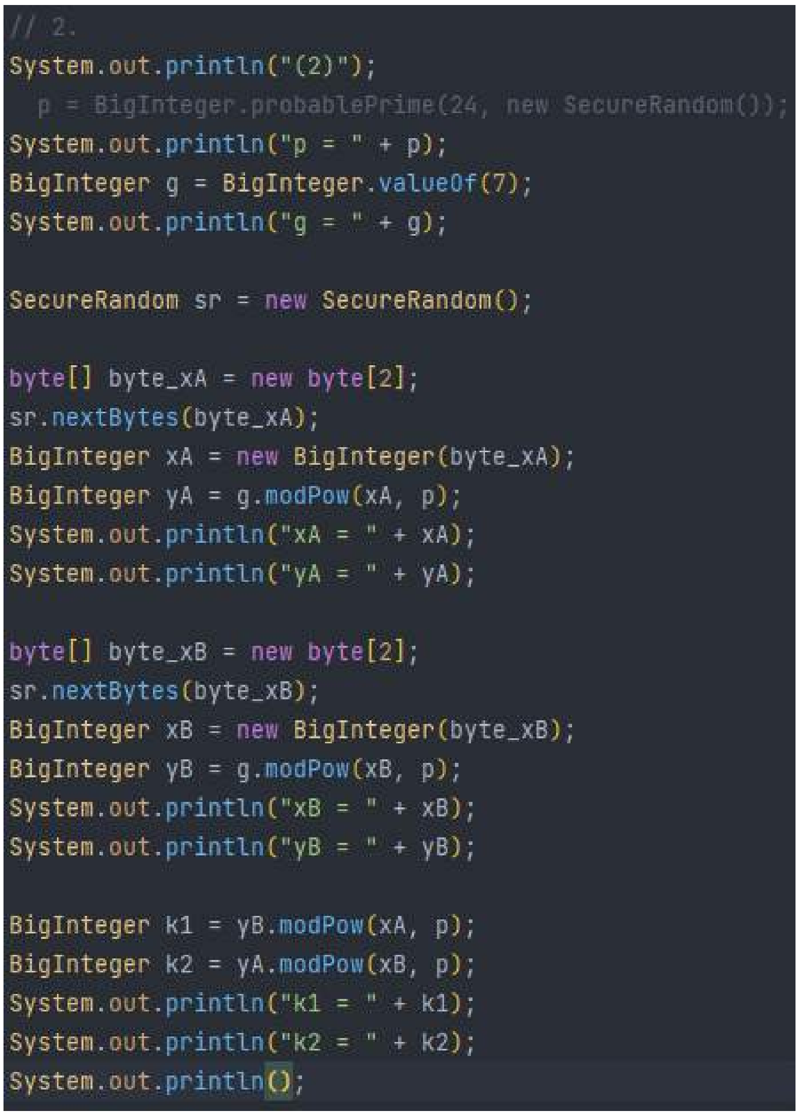
```
(1)
[7, 11, 17, 21, 26, 28, 29, 33, 38, 41, 43, 44, 46, 47, 51, 61, 63, 65, 68, 70,
```

```
[7, 11, 17, 21, 26, 28, 29, 33, 38, 41, 43, 44, 46, 47, 51, 61, 63, 65, 68, 70, 78,
84, 87, 95, 98, 99, 103, 104, 106, 110, 112, 114, 115, 116, 118, 123, 127, 129,
132, 138, 141, 142, 143, 146, 152, 153, 154, 157, 158, 163, 164, 170, 172, 175,
176, 178, 179, 181, 183, 184, 188, 189, 191, 195, 202, 204, 210, 211, 214, 217,
233, 234, 239, 241, 242, 244, 245, 251, 260, 261, 262, 265, 271, 272, 273, 274,
275, 278, 280, 281, 285, 290, 293, 294, 295, 297, 298, 302, 309, 312, 317, 318,
323, 330, 336, 338, 341, 342, 345, 346, 347, 348, 354, 355, 364, 365, 367, 369,
371, 380, 381, 383, 385, 387, 389, 391, 392, 395, 396, 410, 412, 413, 414, 416,
419, 423, 424, 425, 426, 429, 430, 433, 438, 440, 445, 446, 448, 454, 456, 459,
460, 462, 463, 464, 469, 470, 471, 472, 474, 479, 481, 487, 489, 492, 505, 508,
510, 516, 518, 523, 525, 526, 527, 528, 533, 534, 535, 537, 538, 541, 543, 549,
551, 552, 557, 559, 564, 567, 568, 571, 572, 573, 574, 578, 581, 583, 584, 585,
```

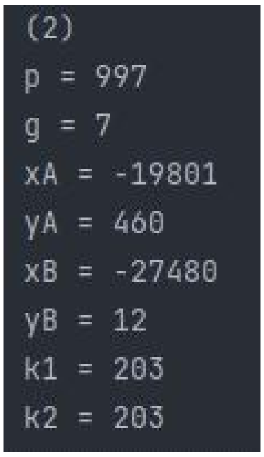
587, 601, 602, 605, 606, 608, 610, 612, 614, 616, 617, 626, 628, 630, 632, 633, 642, 643, 649, 650, 651, 652, 655, 656, 659, 661, 667, 674, 679, 680, 685, 688, 695, 699, 700, 702, 703, 704, 707, 712, 716, 717, 719, 722, 723, 724, 725, 726, 732, 735, 736, 737, 746, 752, 753, 755, 756, 758, 763, 764, 780, 783, 786, 787, 793, 795, 802, 806, 808, 809, 813, 814, 816, 818, 819, 821, 822, 825, 827, 833, 834, 839, 840, 843, 844, 845, 851, 854, 855, 856, 859, 865, 868, 870, 874, 879, 881, 882, 883, 885, 887, 891, 893, 894, 898, 899, 902, 910, 913, 919, 927, 929, 932, 934, 936, 946, 950, 951, 953, 954, 956, 959, 964, 968, 969, 971, 976, 980, 986, 990]

[과제-2]B - (2)

위에서 구한 생성자들 중에 1개를 g로 선정하고, 임의의 y를 선정하여(y≠1) 전수조사를 통해 $y = g^x \bmod p$를 만족하는 x 구하기.



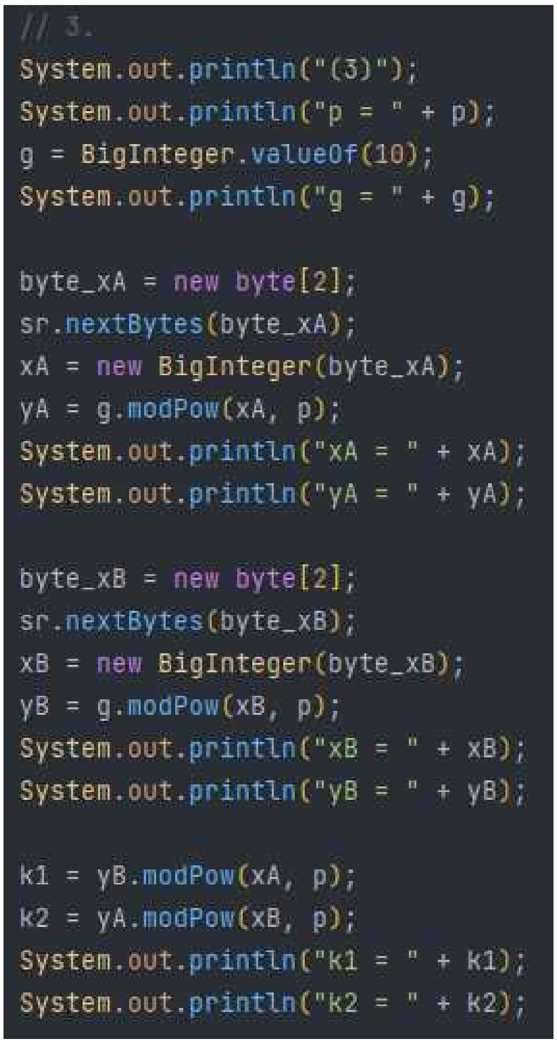
```java
// 2.
System.out.println("(2)");
  p = BigInteger.probablePrime(24, new SecureRandom());
System.out.println("p = " + p);
BigInteger g = BigInteger.valueOf(7);
System.out.println("g = " + g);

SecureRandom sr = new SecureRandom();

byte[] byte_xA = new byte[2];
sr.nextBytes(byte_xA);
BigInteger xA = new BigInteger(byte_xA);
BigInteger yA = g.modPow(xA, p);
System.out.println("xA = " + xA);
System.out.println("yA = " + yA);

byte[] byte_xB = new byte[2];
sr.nextBytes(byte_xB);
BigInteger xB = new BigInteger(byte_xB);
BigInteger yB = g.modPow(xB, p);
System.out.println("xB = " + xB);
System.out.println("yB = " + yB);

BigInteger k1 = yB.modPow(xA, p);
BigInteger k2 = yA.modPow(xB, p);
System.out.println("k1 = " + k1);
System.out.println("k2 = " + k2);
System.out.println();
```



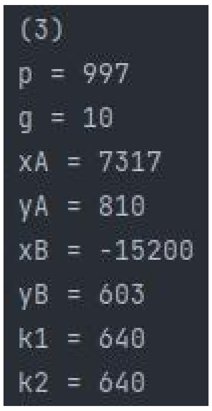
```
(2)
p = 997
g = 7
xA = -19801
yA = 460
xB = -27480
yB = 12
k1 = 203
k2 = 203
```

[과제-2]B - (3)

생성자가 아닌 g를 선정하고, 임의의 y를 선정하여(y≠1) 전수조사를 통해 $y = g^x \bmod p$를 만족하는 x 구하기.



```
// 3.
System.out.println("(3)");
System.out.println("p = " + p);
g = BigInteger.valueOf(10);
System.out.println("g = " + g);

byte_xA = new byte[2];
sr.nextBytes(byte_xA);
xA = new BigInteger(byte_xA);
yA = g.modPow(xA, p);
System.out.println("xA = " + xA);
System.out.println("yA = " + yA);

byte_xB = new byte[2];
sr.nextBytes(byte_xB);
xB = new BigInteger(byte_xB);
yB = g.modPow(xB, p);
System.out.println("xB = " + xB);
System.out.println("yB = " + yB);

k1 = yB.modPow(xA, p);
k2 = yA.modPow(xB, p);
System.out.println("k1 = " + k1);
System.out.println("k2 = " + k2);
```



```
(3)
p = 997
g = 10
xA = 7317
yA = 810
xB = -15200
yB = 603
k1 = 640
k2 = 640
```