```
public class ElGamal{




    public static void main(String args[])
    {    // This code demonstrates the  El Gamal algorithm
        // notation as in slides
        // prime  p = 467 , g = 2
        // Let Alice's private key be 153
        // we need to work out Alice's public key
        // Alice public key = g^{153} mod 467
        int p = 467, g = 2, Alice_Pri=153;
        int Alice_Pub=1;

        // we want 2*2*2*2*2* ..... 153 times mod 467
        for(int j=1; j <=Alice_Pri;j++ )
            {
                Alice_Pub = (g*Alice_Pub) % p;
            }
        System.out.println("Alice's public key is " + Alice_Pub);
        // Bob wants to send Alice the message M=331
        // and let  k = 197
        // He computes c_1 and c_2
        // what are the values?
        int c_1=1, c_2=1, M=331;
        // refer to notes 1.1.3 and eqn 2
        // c_1 = g^k mod p and c_2=  M * (Alice_Pub)^k mod p
        // Bob picks k
                int k =197;
                // c_1 = g^k mod p
        for (int j = 1;j <=k;j++)
            {
                c_1 = g*c_1 % p;
            }
            System.out.println("c_1= " + c_1);
            //c_2=  M * (Alice_Pub)^k mod p
          for (int j = 1;j <=k;j++)
            {
                c_2 = (Alice_Pub*c_2) % p;
            }
          c_2 = (M * c_2) % p;
                    System.out.println("c_2= " + c_2);
          // Bob sends c_1 and c_2 to Alice
           // ALice wishes  to read Bob's message,
           // to this goal Alice  evaluates x = (c_1)^{Alice_Pri} mod p
          // Note Eve does not know Alice_Pri
                    int x=1;
          for (int j = 1;j <=Alice_Pri;j++)
            {
                x = (c_1*x) % p;   // 1.3 eqn 3
            }
          System.out.println("x= " + x);

          // seek x inverse,  1.3  eqn 3  - second part
          // there are  quicker ways of doing this
          // note for large p this is laborious
          // there  are quicker algorithms
          // but we keep it simple
          int xinv=1;
           for (int j = 1;j <=p-1;j++)
               {
                   if ((j*x)%p==1)
                       {
                           xinv=j;
                        System.out.println("xinv= " + xinv);
                       }
               }
           //
           // Finally
           int M2 = (c_2*xinv) % p;
                        System.out.println("M2= " + M2);
                        // this needs to equal M


    }

}
```