# Lab8 Report

1. Name: 林靖昀 Student ID: B12902116

2. The plaintext you get from ciphertext in course ppt page 64

   With:
   ```
   phi = (p - 1) * (q - 1)
   d = inverse(e, phi)
   m = pow(c, d, p * q)
   ```

   Ans: 557744663355446577768879988

3. The plaintext you get from ciphertext in course ppt page 65

   From `openssl rsa -inform PEM -pubin -in public-key.pem -text -noout`
   we can get the n and e, n is small enough to factor on factordb, thus we get:

   'b'\x02yq\x96\xdaU\xa2\x077\xa0f\xfa\xe1\x7f\x0f\x0c\xbc\xddA\xac\xf8L\x91\xf3:\xc3Mq\xec\xc7v\x14\xef#\xfc\xd5<`\x98\xaa\xe4\xdd\xb8B\x9fK\xca?\xe8\xf7\x9cQ\x0f\x91\x80\xe5Q\xba\xaa\x1ak\x00FLAG_IS_WeAK_rSA\n''

   Ans:: FLAG_IS_WeAK_rSA

4. The plaintext you get from picture in course ppt page 77 exercise 4

   Extract all filter bits from every chunk (I used the scripts from:
   https://github.com/PotatoKingTheVII/png-filter-steg)
   Group bits by 8 and reverse them, then decode

   Ans: DrgnS{WhenYouGazeIntoThePNGThePNGAlsoGazezIntoYou}

5. The plaintext you get from picture in course ppt page 77 exercise 5

   Using stegsolve, with random color map, we can see the flag

   Ans: pctf{keep_doge_alive_2014}

6. The plaintext you get from picture in course ppt page 77 exercise 6

   After extracting each LSB from each pixel in the png and decoding the binary, we see "Rar!" at the start.
   After writing the binary data to a file and trying to unrar it, we see that it needs a password.
   Using john the ripper, we get the password "brute".
   After unraring, we get the flag.txt:

   Ans: {LSB_is_ubiquitous}