**超級可以被破解的超安全加密系統**

**RSA algorithm:**

1. 選擇兩個大質數 P, Q, 讓 N = P\*Q，且P不等於Q
2. 再計算一個Φ(N), Φ(N) 是用於計算可以使用的 加密金鑰 **e**
3. 選擇一個加密金鑰e，(e, N)作為公開金鑰, e大於1且小於Φ(N)，然後與N, Φ(N)互質，跟N, Φ(N)互質是為了取得一個可以循環的模數
4. 計算解密金鑰d, ，同等於ed (mod) = 1, 意同 d 為 e 的模反元素 使其 訊息 M 經過計算 Me mod N = C , 會獲得 密文 C
5. Me (mod N) = C, Cd (mod N) = M

**PROC:**

RSAlgorithm PROTO, Mt:DWORD, E:DWORD, N:DWORD

參數 Mt (message) 是要加解密的訊息

E (Cryptographic key) 用於加解密的金鑰

N 是很難分解的大數，用於計算加密循環的被模數

power PROTO , a:DWORD, n:DWORD

eax回傳 a的n次方

readInteger PROTO, msg:PTR BYTE, target:PTR DWORD

輸出msg讀取一個整數，並寫入target的function

printMessage PROTO, msg:PTR BYTE, value:DWORD

輸出message並輸出value

printlnMessage PROTO, msg:PTR BYTE, value:DWORD

printMessage並換行

setup PROTO, P:PTR DWORD, Q:PTR DWORD, N:PTR DWORD, PN:PTR DWORD, E:PTR DWORD, D:PTR DWORD

公私鑰生成的初始化function

參數 P, Q, N , PN, E, D只是要儲存的記憶體位置

coprimeTest PROTO, num:DWORD, N:DWORD, PN:DWORD

用於測試num是否與N, PN互質

generatePrivateKey PROTO, E:DWORD, N:DWORD, D:PTR DWORD

透過加密金鑰e生成解密金鑰d

readFileSync PROTO, bufPTR:PTR DWORD, bufSize:DWORD

讀取檔案到buffer pointer指向的記憶體上

encrypteBuffer PROTO, bufPTR:PTR DWORD, bufSize:DWORD, target:PTR DWORD

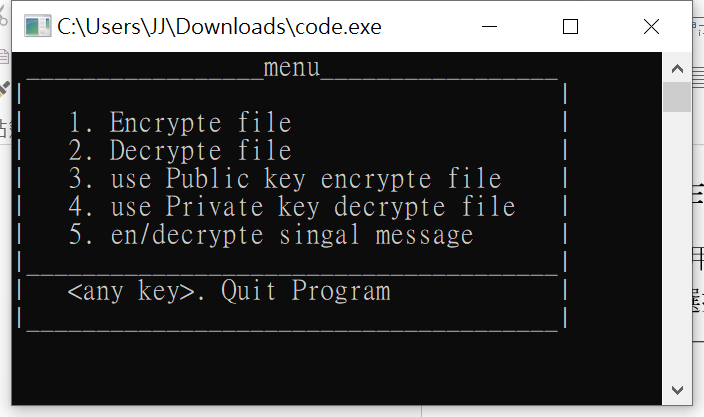
加密指定記憶體上的資料，並將計算結果寫入target buffer的記憶體上

decrypteBuffer PROTO, bufPTR:PTR DWORD, bufSize:DWORD, target:PTR DWORD

由於加密的buffer與解密用的buffer size不同，因此將encrypte, decrypte分開實作

解密記憶體上指定的資料，並將計算結果寫入target buffer上

**操作流程:**



可使用的功能說明

1. 選擇要透過哪種加密方法
2. 加密檔案, 此功能會需要選擇 p, q, e 並自動生成d 再加密檔案, 可透過計算的 (d, N) 再利用 **功能四** 解密檔案
3. 解密檔案, 手動選擇 p, q, e 並自動計算, 之後可透過 **功能三** 直接使用 (e, N) 直接加密檔案
4. 透過加密金鑰直接加密檔案 (不必選擇 p, q也不會自動生成 d)
5. 透過解密金鑰直接解密檔案 (不必選擇 p, q 也不會自動生成 e)
6. 選擇 p, q, n 自動加解密一個輸入的 integer

**功能:**

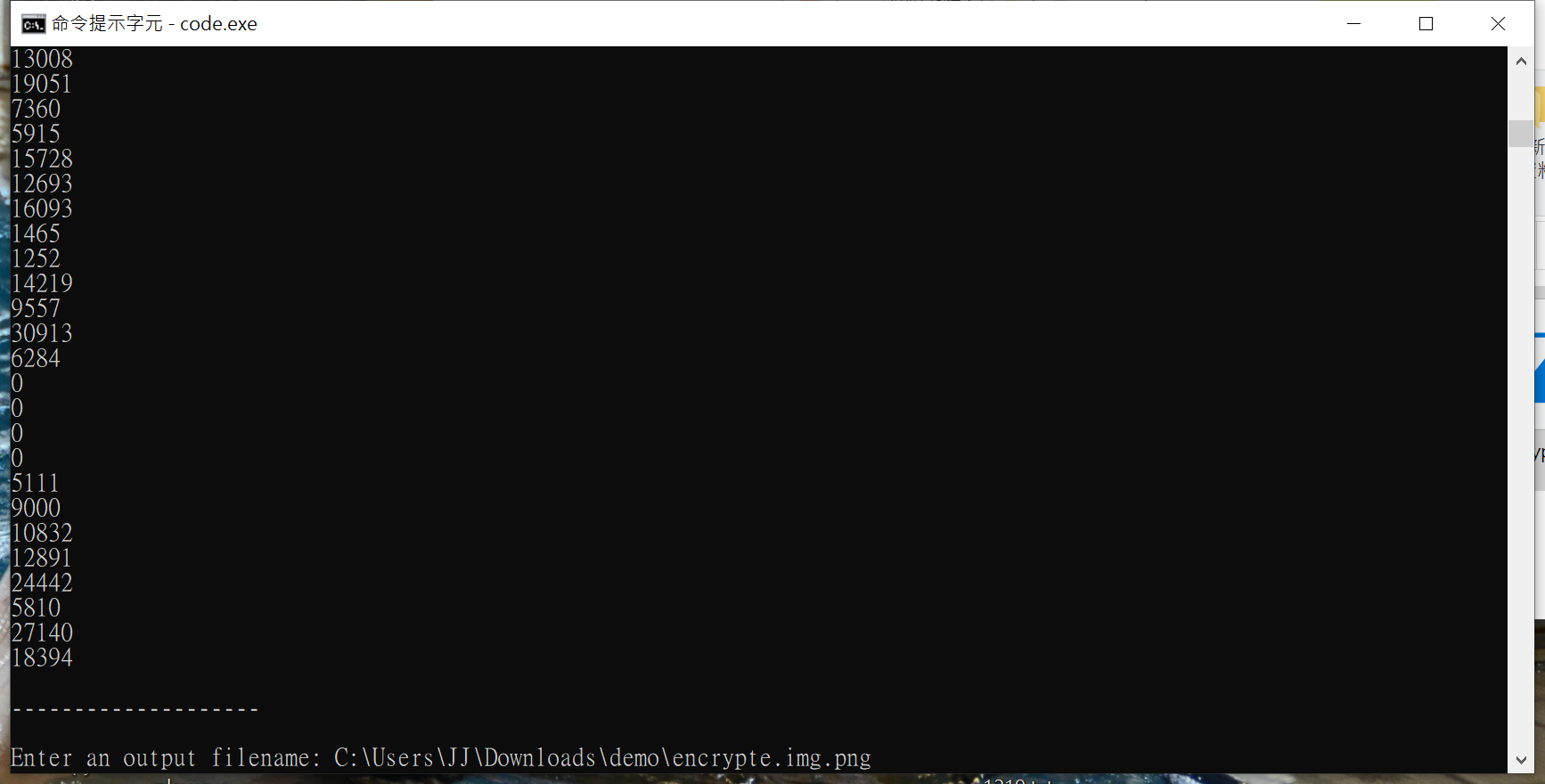
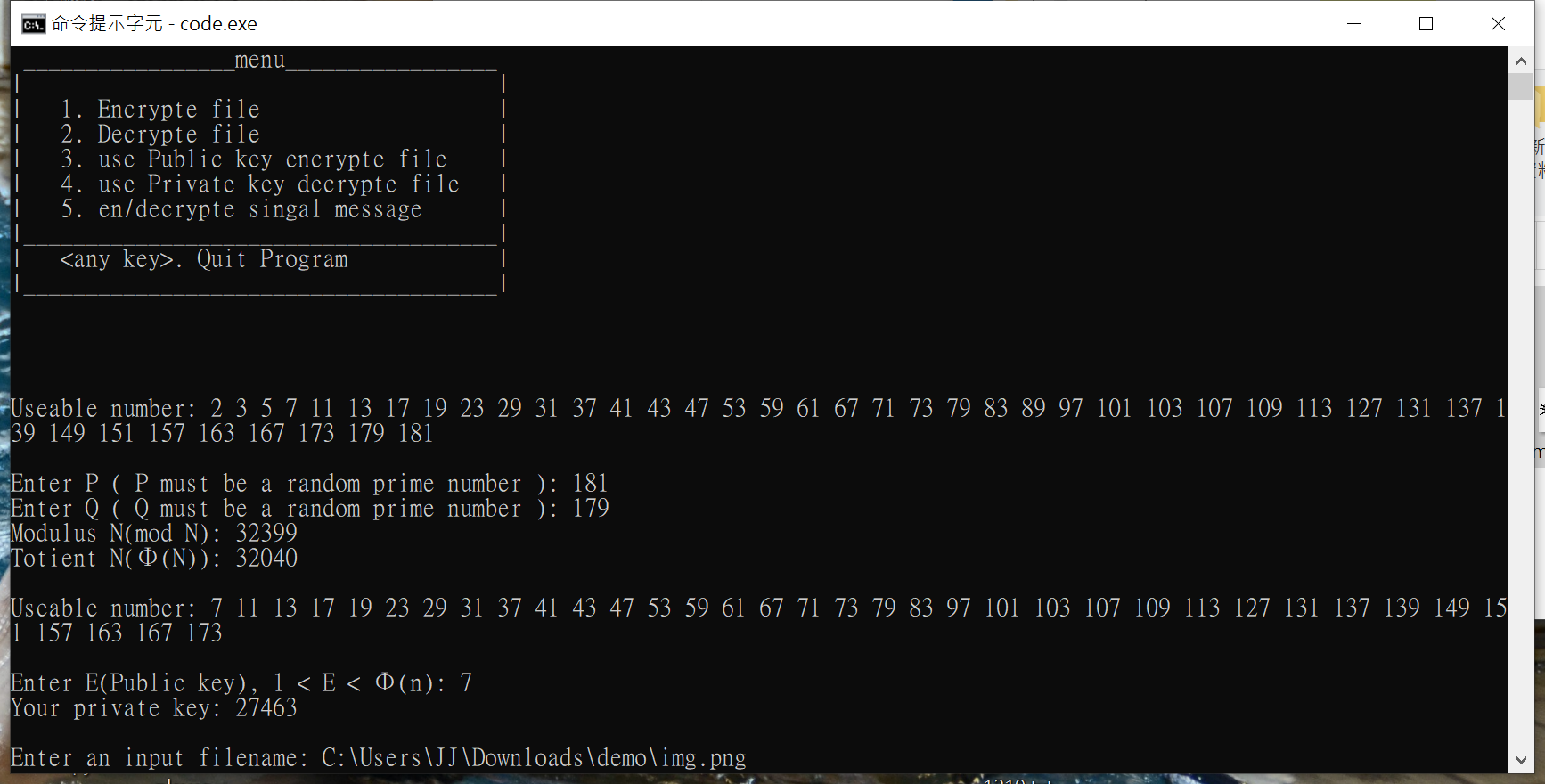
1. 生成 e, d, N 並加密檔案

1. 隨機選擇列表提供的兩個質數p, q , 且 p不等於 q

1. 選擇列表提供的隨意加密金鑰e

選擇e之後會自動計算解密金鑰d ; Your private key: **{Integer}**

1. 輸入的路徑 或者 拖曳檔案
2. 輸入要輸出的檔案路徑 或者 拖曳輸出的檔案



1. 生成 e, d, N 並解密檔案

1. 隨機選擇列表提供的兩個質數p, q , 且 p不等於 q

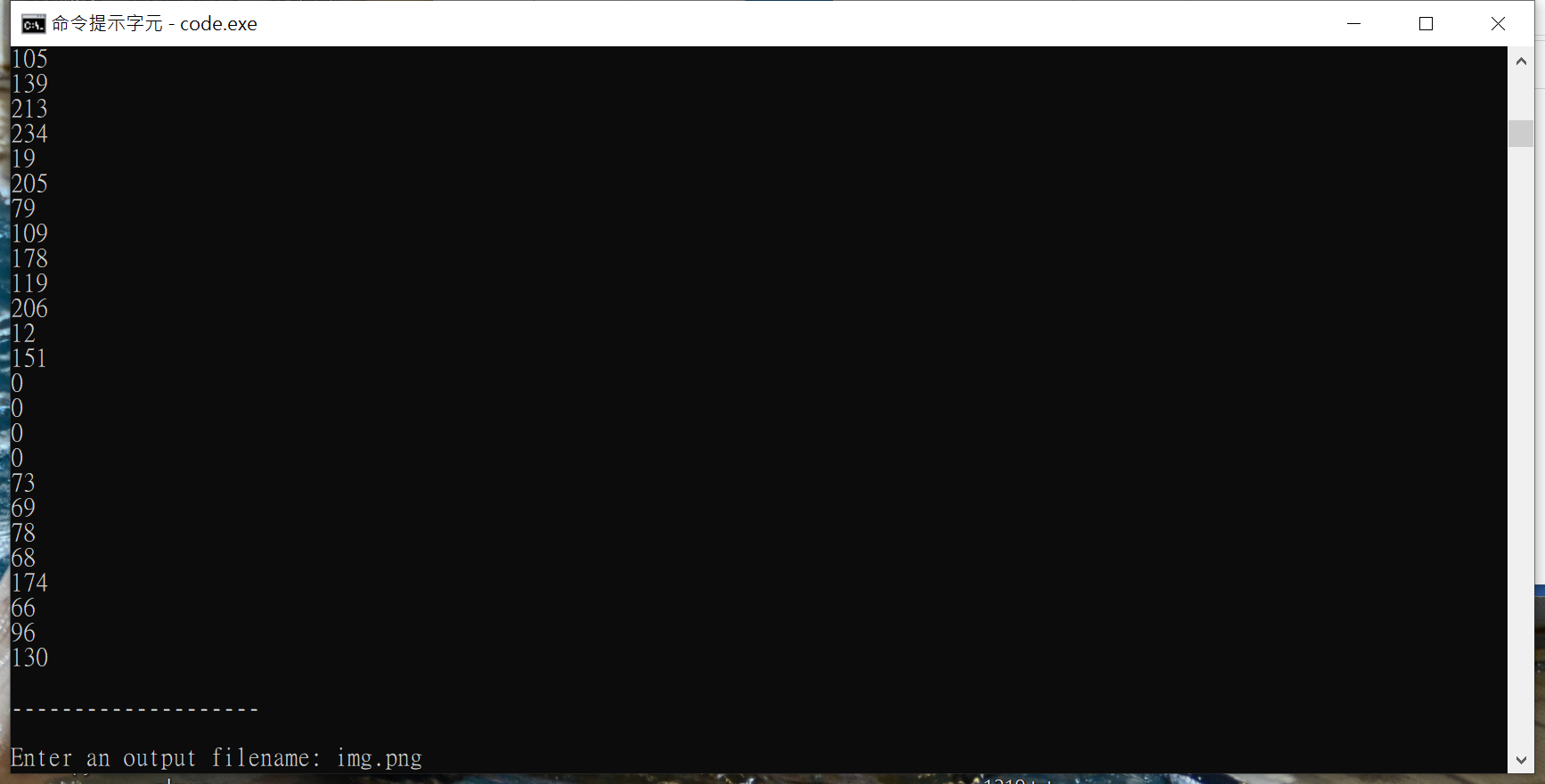
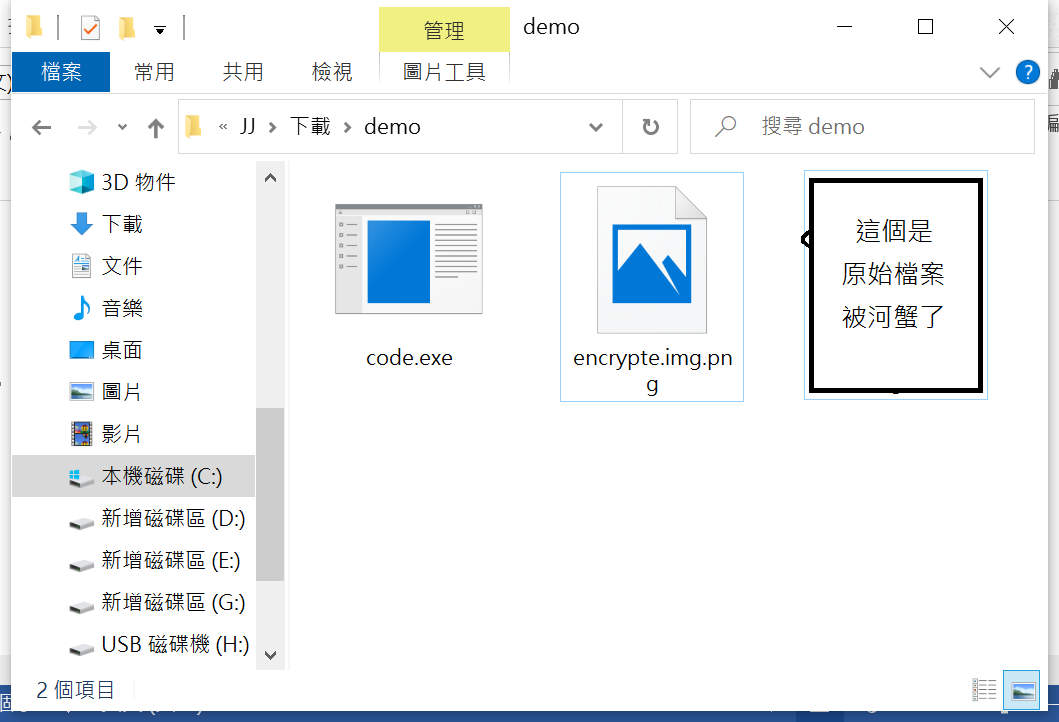
2. 選擇列表提供的隨意加密金鑰e

選擇e之後會自動計算解密金鑰d ; Your private key: **{Integer}**

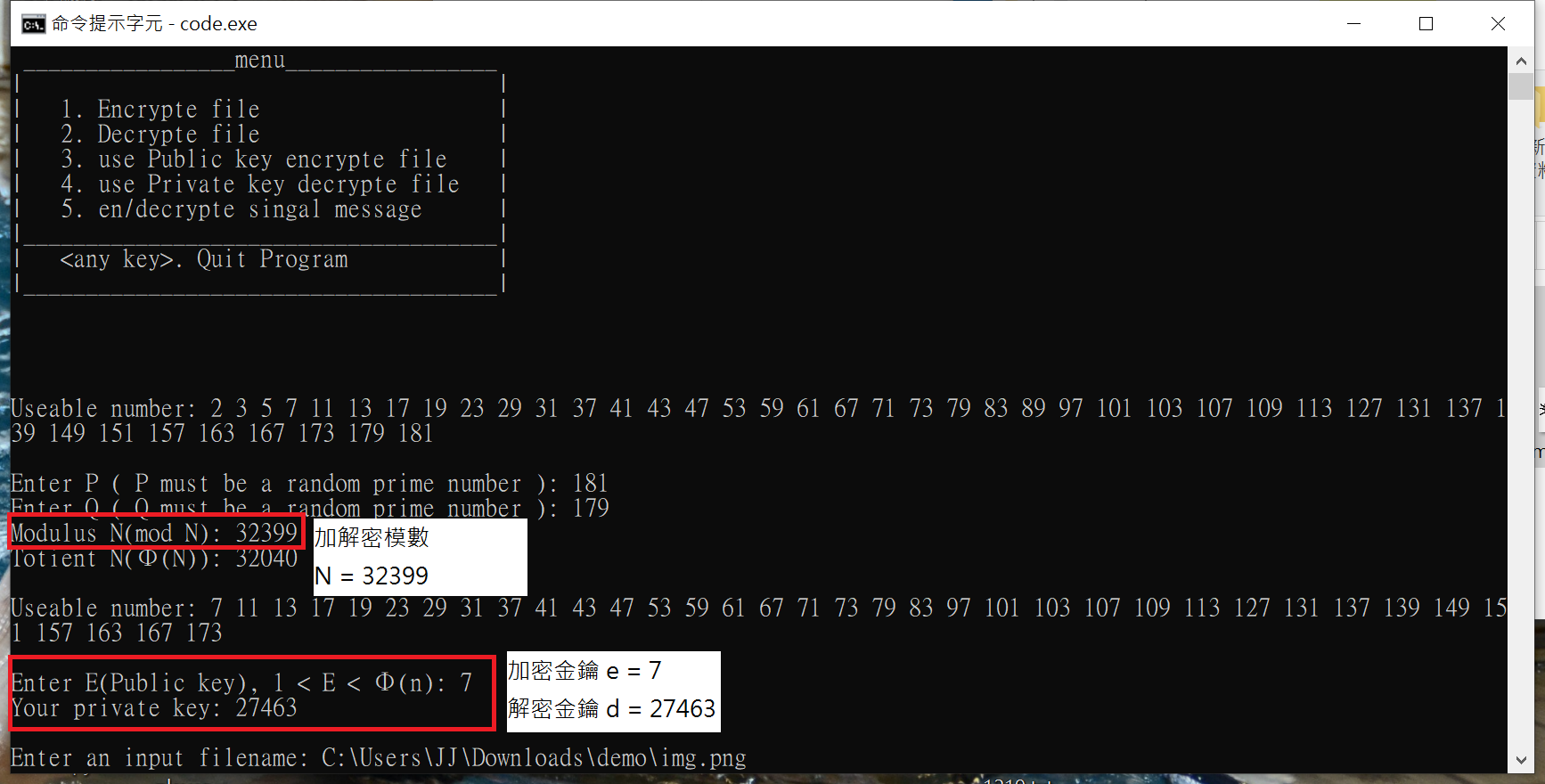
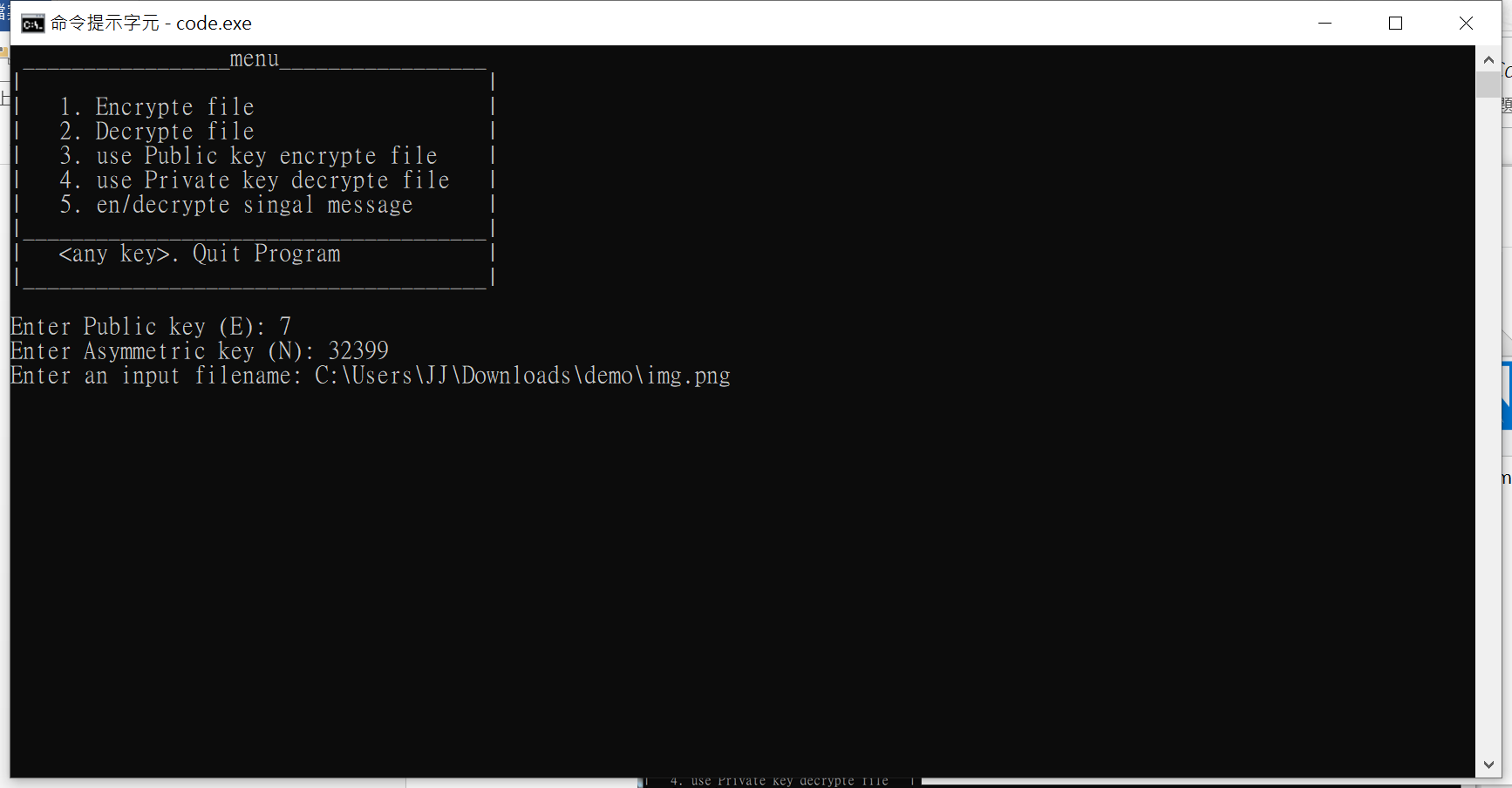
1. 輸入要解密檔案的路徑 或者 拖曳被加密的檔案
2. 輸入要輸出的檔案路徑 或者 拖曳輸出的檔案

[加密的檔案連結](https://github.com/tomorrow-have-advanced-calculus/rsa-assembly/blob/master/encrypted.image.png)

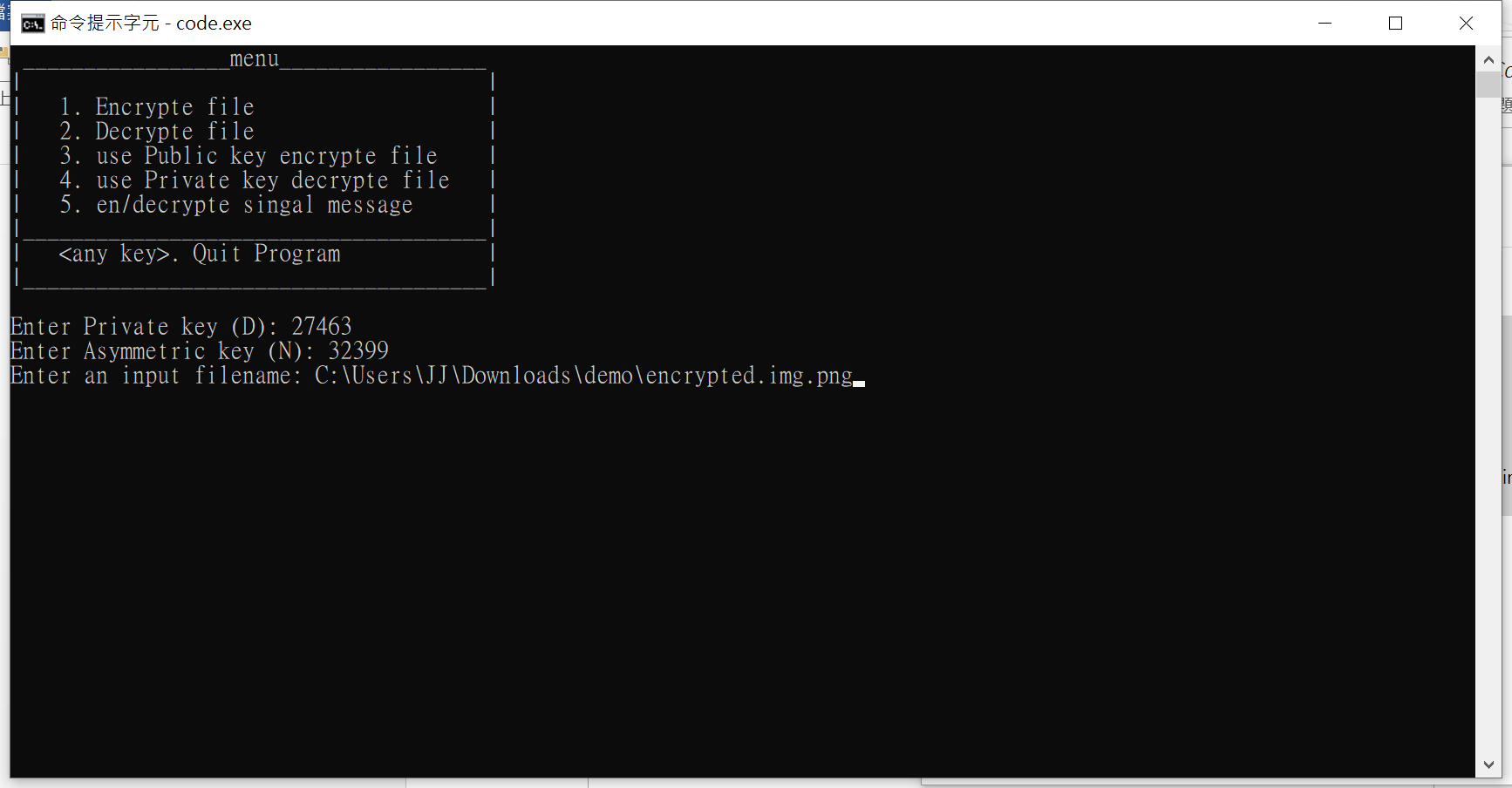
[加解密金鑰 e, d, N](https://github.com/tomorrow-have-advanced-calculus/rsa-assembly/blob/master/key-example)



1. 使用 加密金鑰 加密檔案
2. 記錄金鑰 e, d, N
3. 輸入 e, N
4. 輸入檔案路徑 或者 拖曳檔案到視窗中
5. 輸入想存檔的路徑 或者 拖曳想複寫的檔案



1. 使用 解密金鑰 加密檔案
2. 記錄金鑰 e, d, N
3. 輸入 d, N
4. 輸入檔案路徑 或者 拖曳檔案到視窗中
5. 輸入想存檔的路徑 或者 拖曳想複寫的檔案



1. 加解密單一 數字 (ASCII code)
2. 選擇 p, q
3. 輸入隨意的 e, 紀錄 d
4. 輸入想加密的訊息
5. 紀錄 Result 的數字
6. 重新執行程式
7. 選擇與上次相同的 p, q
8. 輸入上次生成的 d
9. 輸入上次 Result的數字
10. 取得原始結果 Result: 1024

