

MSSV: 19127102

Họ tên: Võ Hoàng Gia Bảo

1. Có 3 file KeyGeneration.py, EncryptMess.py, và DecryptMess.py. 3 file được chạy theo thứ tự đã sắp xếp.

2. Chạy file KeyGeneration.py bằng PyCharm. Source code được sử dụng lại từ bài tập tuần 3, gồm có hàm kiểm gcd bằng bezout, hàm sinh cặp khóa rsa và hàm kiểm tra số nguyên tố lớn rabin-miller ($> 10^5$ chữ số)

P/S: có thể thay đổi độ dài của khóa tại length=...

```
def RSA_key_generation(length=10):      # Ex: 128 -> 1024 bits
    p = q = 0
    while p == 0 or q == 0:
        temp = random.randint(0, 10 ** length)
        if is_prime(temp):
            if p == 0:
                p = temp
            elif len(str(p)) == len(str(temp)):
                q = temp
    n = p*q
    euler = (p-1)*(q-1)
    e = random.randint(2, euler)
    while bezout(e, euler)[0] != 1:
        e = random.randint(2, euler)
    d = bezout(e, euler)[2] % euler
    return e, n, d
```

Sau khi chạy hệ thống sẽ yêu cầu nhập path để lưu trữ khóa public tên là rsa_pub.txt và khóa private rsa.txt

```
C:\Users\USER\AppData\Local\Programs\Python\Python39\python.exe D:/ASUS/Documents/  
Input path to store keys: D:\\ASUS\\Documents\\Homework\\MH-MM\\3\\19127102  
  
Process finished with exit code 0
```

Lưu ý: Phải nhập đúng cú pháp

VD: C:\\Users\\USER\\Desktop

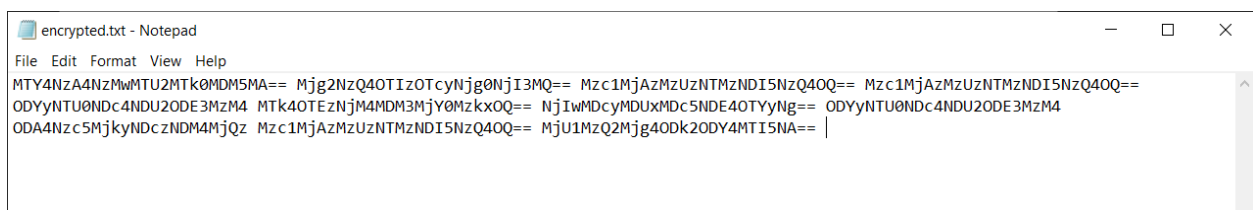
3. Chạy file EncryptMess.py (file EncryptMess.py không nhất thiết phải nằm cùng folder với các file khác) bằng PyCharm

Sau khi chạy hệ thống sẽ yêu cầu nhập tên / đường dẫn file txt chứa đoạn văn bản cần mã hóa và file txt của **khóa công khai**.

VD: Trong folder bài tập đã có file TestInput.txt với đoạn “Hello World”

```
C:\Users\USER\AppData\Local\Programs\Python\Python39  
Input file name to encrypt: TestInput.txt  
Input public key file name to encrypt: rsa_pub.txt  
  
Process finished with exit code 0
```

Sau đó trong folder có chứa file py sẽ có thêm 1 file encrypt.txt



```
encrypted.txt - Notepad  
File Edit Format View Help  
MTY4NzA4NmMwMTU2MTk0MDM5MA== Mjg2NzQ4OTIzOTcyNjg0NjI3MQ== Mzc1MjAzMzUzNTMzNDI5NzQ4OQ== Mzc1MjAzMzUzNTMzNDI5NzQ4OQ==  
ODYyNTU0NDc4NDU2ODE3MzM4 MTK4OTEzNjM4MDM3MjY0MzIxOQ== NjIwMDcyMDUxMDc5NDE4OTYyNg== ODYyNTU0NDc4NDU2ODE3MzM4  
ODA4Nzc5MjkyNDc4NDM4MjQz Mzc1MjAzMzUzNTMzNDI5NzQ4OQ== MjU1MzQ2Mjg4ODk2ODY4MTI5NA== |
```

Hệ thống sẽ mã hóa **từng ký tự**, mỗi ký tự là 1 byte, sau đó encode base64 để lưu chuỗi mã hóa vào file encrypt.txt. Công thức mã hóa:

$$c \equiv m^e \pmod n$$

```
temp = str(pow(m, int(pub_key[0]), int(pub_key[1])))
```

Với pub_key [0] là e, pub_key [1] là n để suy ra được c

For char in plaintext:

m <= ascii-value (char)

encrypted append (m ** e % n)

4. Chạy file DecryptMess.py (file DecryptMess.py không nhất thiết phải nằm cùng folder với các file khác) bằng PyCharm

Sau khi chạy hệ thống sẽ yêu cầu nhập tên / đường dẫn file txt chứa đoạn văn bản cần giải mã và file txt của khóa bí mật.

VD: Lấy lại file txt đã mã hóa trước đó là encrypted.txt

```
C:\Users\USER\AppData\Local\Programs\Python\Python34\
Input file name to decrypt: encrypted.txt
Input private key file name to decrypt: rsa.txt

Process finished with exit code 0
```

Hệ thống sẽ decode base64 trước khi giải mã từng phần trong file encrypted.txt, sau đó lưu chuỗi giải mã vào file decrypt.txt. Công thức giải mã:

$$c^d \equiv m^{de} \equiv m \pmod n$$

```
temp = base64.b64decode(i.encode('ascii')).decode('ascii')
decrypted += chr(pow(int(temp), int(pri_key[0]), int(pri_key[1])))
```

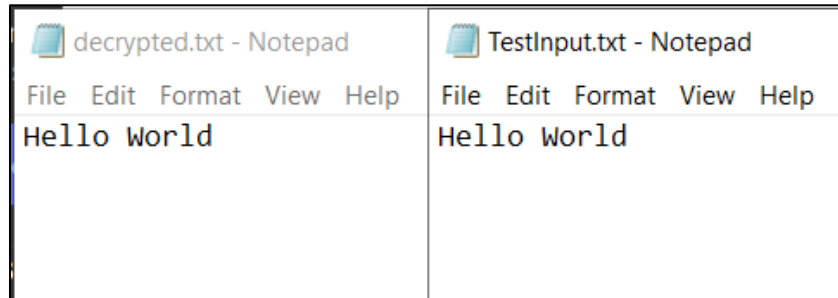
Với temp là c, pri_key [0] là d, pri_key [1] là n để suy ra lại được m

For part in ciphertext:

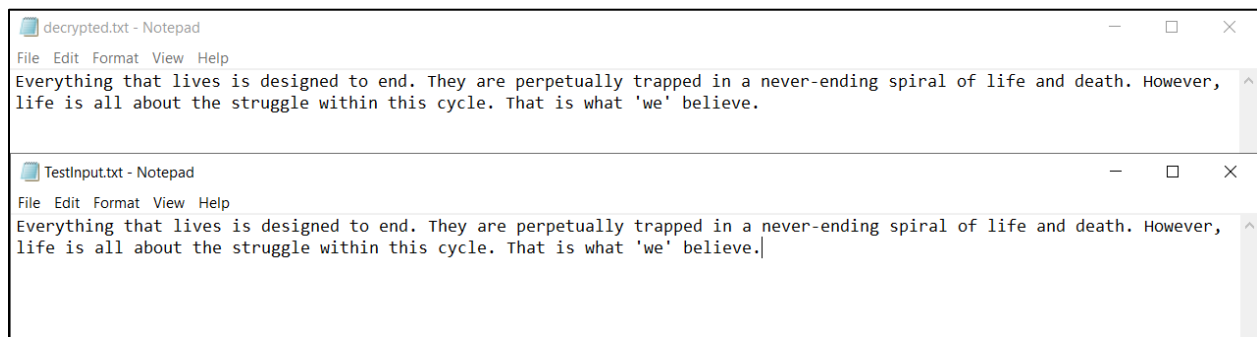
if part is not null

decrypted append (part ** d % n)

Kết quả file gốc và file được mã hóa:



VD khác:



Nguồn tham khảo: [Hệ mã hóa RSA và chữ ký số \(viblo.asia\)](https://viblo.asia)