HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN

BÁO CÁO

Bài 16: Lập trình thuật toán mật mã học

Giảng viên hướng dẫn: Vũ Minh Mạnh

Sinh viên thực hiện: Nguyễn Quốc Vượng

Mã sinh viên: B21DCAT227

Lớp: D21CQAT03-B

Hà Nội, 2023

Môn học Thực tập cơ sở

Bài 16: Lập trình thuật toán mật mã học

1.1 Mục đích

Sinh viên tìm hiểu một giải thuật mã hóa phổ biến và lập trình được chương trình mã hóa và giải mã sử dụng ngôn ngữ lập trình phổ biến như C/C++/Python/Java, đáp ứng chạy được với số lớn.

1.2 Nội dung thực hành

1.2.1 Tìm hiểu lý thuyết

- Tìm hiểu về lập trình số lớn với các phép toán cơ bản

Lý thuyết số lớn là một lĩnh vực trong toán học nghiên cứu về tính chất và phép toán của các số nguyên lớn, đặc biệt là các số nguyên tố và phân tích của chúng. Lý thuyết số lớn đóng vai trò quan trọng trong nhiều lĩnh vực khác nhau, bao gồm mật mã, lý thuyết đồng hồ, và khoa học máy tính.

Một số vấn đề quan trọng trong lý thuyết số lớn bao gồm:

1. Phân tích số nguyên tố: Nghiên cứu cách phân tích một số lớn thành các thành phần nguyên tố của nó.
2. Định lý số nguyên tố: Những kết quả quan trọng về phân phối của số nguyên tố trong các dãy số.
3. Phân tích mã hóa: Sử dụng tính chất của các số nguyên lớn để tạo ra các thuật toán mã hóa và giải mã.
4. Công nghệ mật mã: Ứng dụng lý thuyết số lớn trong việc phát triển các hệ thống mật mã bảo mật.
5. Các vấn đề liên quan đến chuỗi số: Nghiên cứu các đặc điểm của các chuỗi số như chuỗi Fibonacci, chuỗi nguyên tố, và các mẫu số khác.

Lý thuyết số lớn cũng liên quan chặt chẽ đến các lĩnh vực khác của toán học như đại số và hình học.

- Tìm hiểu về giải thuật mật mã khóa công khai RSA

➢ **Khởi tạo**

- **Chọn hai số nguyên tố lớn**: Chọn hai số nguyên tố lớn p và q. Độ dài của các số nguyên tố này thường rất lớn, để tăng tính an toàn của thuật toán.
- **Tính n**: Tính n bằng cách nhân hai số nguyên tố lớn p và q: n = p * q. N là thành phần chính của các khóa trong thuật toán RSA.

- **Chọn số nguyên e**: Chọn một số nguyên e sao cho $1 < e < \varphi(n)$, với $\varphi(n)$ là hàm số Euler của n, tức là số các số nguyên tố cùng nhau với n trong khoảng từ 1 đến n. e thường được chọn là một số nguyên tố và cùng nhau với $\varphi(n)$.
- **Tính d**: Tìm số nguyên d sao cho $(d * e) \bmod \varphi(n) = 1$. d là nghịch đảo modular của e modulo $\varphi(n)$.

➢ **Mã hóa**

- **Khóa công khai**: Cặp (e, n) là khóa công khai, được chia sẻ với bất kỳ ai muốn gửi tin nhắn cho chủ sở hữu của khóa.
- **Mã hóa tin nhắn**: Mỗi ký tự hoặc khối văn bản sẽ được biến đổi thành một số (thường là theo mã ASCII) và sau đó mã hóa bằng cách tính c^e mod n, với c là số đã được biểu diễn của văn bản.

➢ **Giải mã**

- **Khóa bí mật**: Cặp (d, n) là khóa bí mật, được giữ bí mật bởi chủ sở hữu của khóa. Khóa bí mật này được sử dụng để giải mã tin nhắn đã được mã hóa.
- **Giải mã tin nhắn**: Sử dụng khóa bí mật (d, n) để giải mã số đã được mã hóa c. Mỗi số đã được mã hóa sẽ được giải mã bằng cách tính c^d mod n.

1.2.2 Chuẩn bị môi trường

- Môi trường lập trình theo mong muốn.

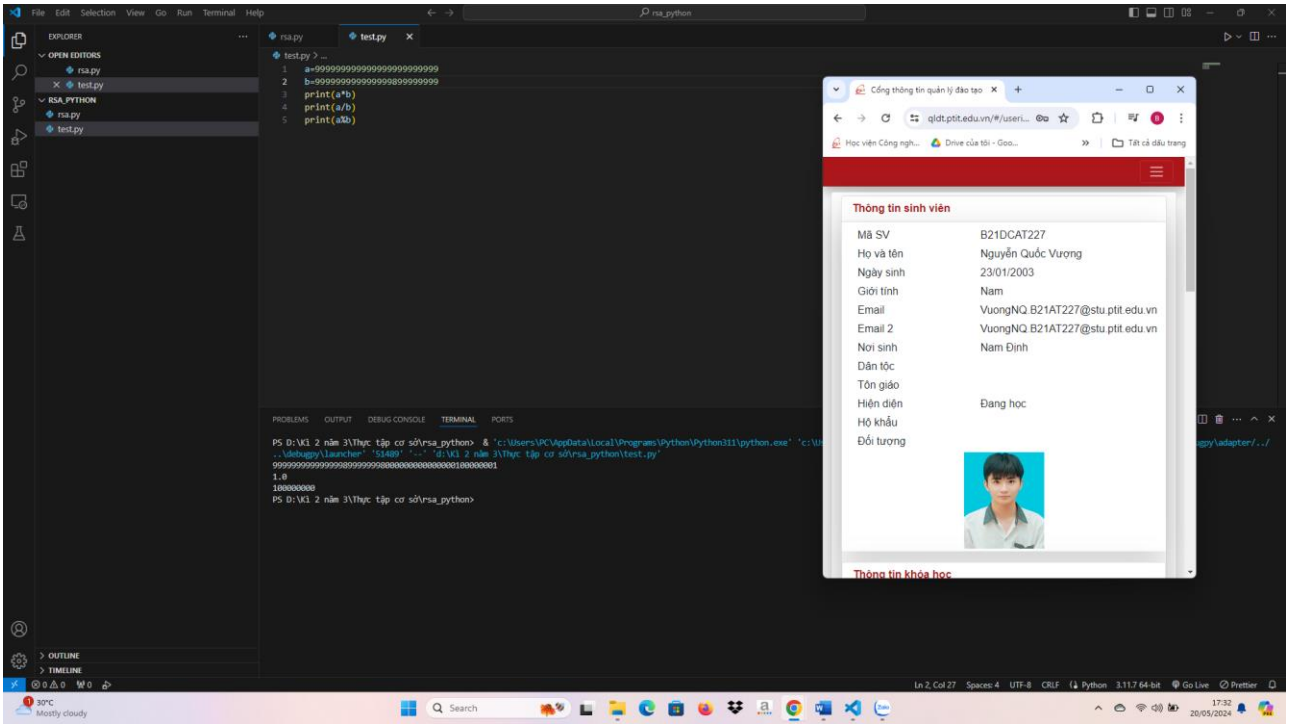
1.2.3 Các bước thực hiện và kết quả cần đạt

a) Các bước thực hiện

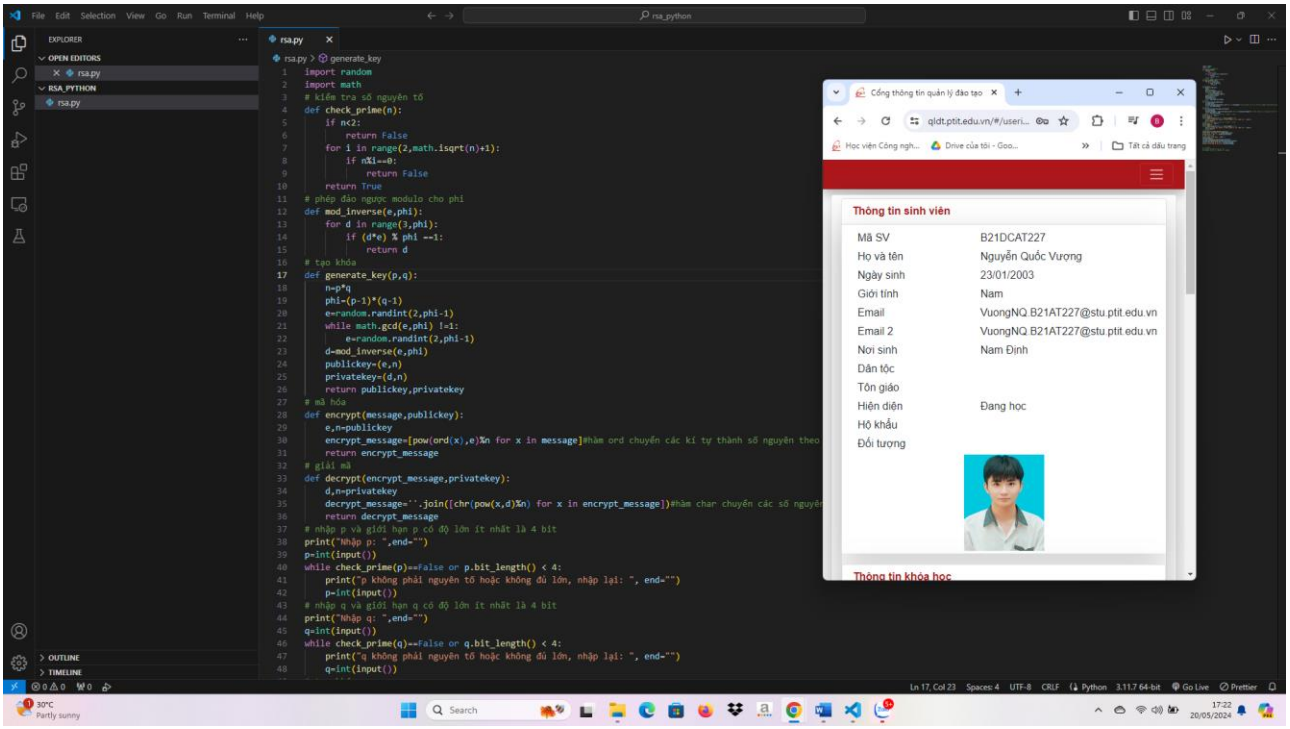
- Lập trình thư viện số lớn với các phép toán cơ bản để sử dụng trong giải thuật mã hóa/giải mã RSA

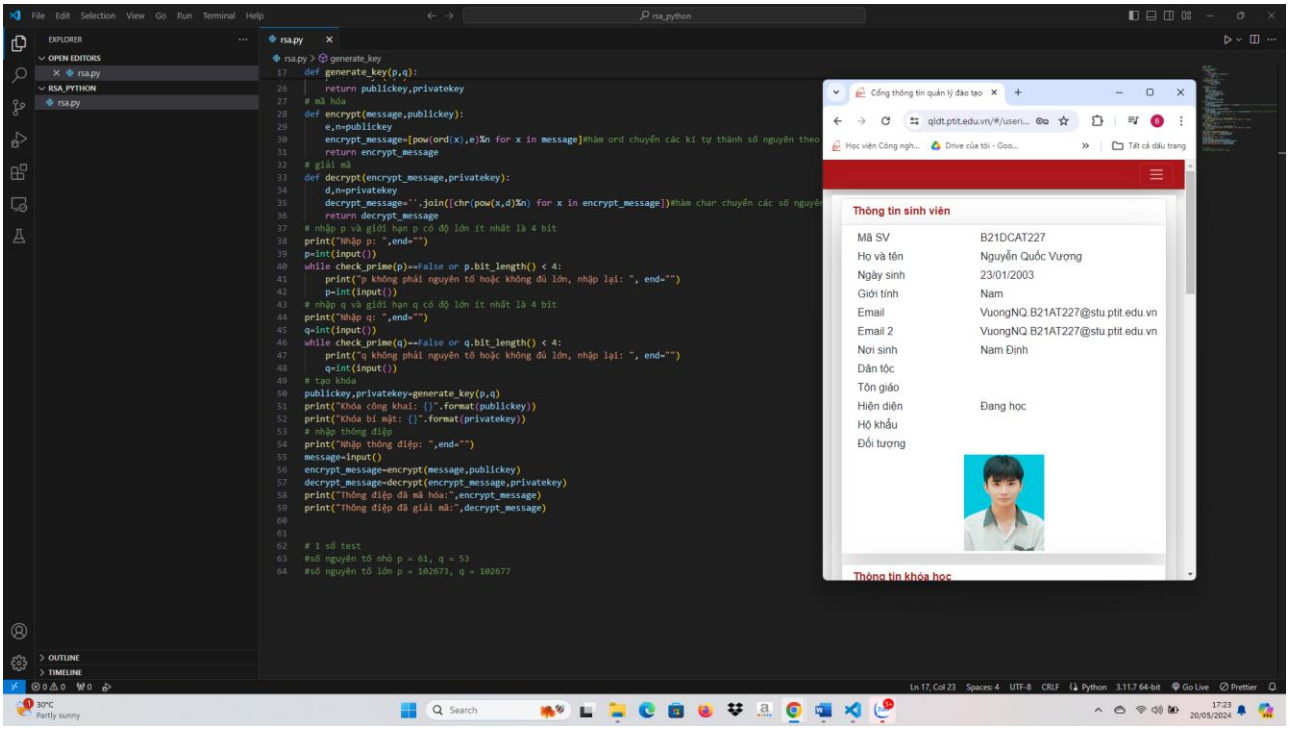
Trong python, các phép toán với số nguyên lớn có thể làm việc với kiểu dữ liệu int

- Thử nghiệm chứng minh thư viện hoạt động tốt với các ví dụ phép toán cho số lớn

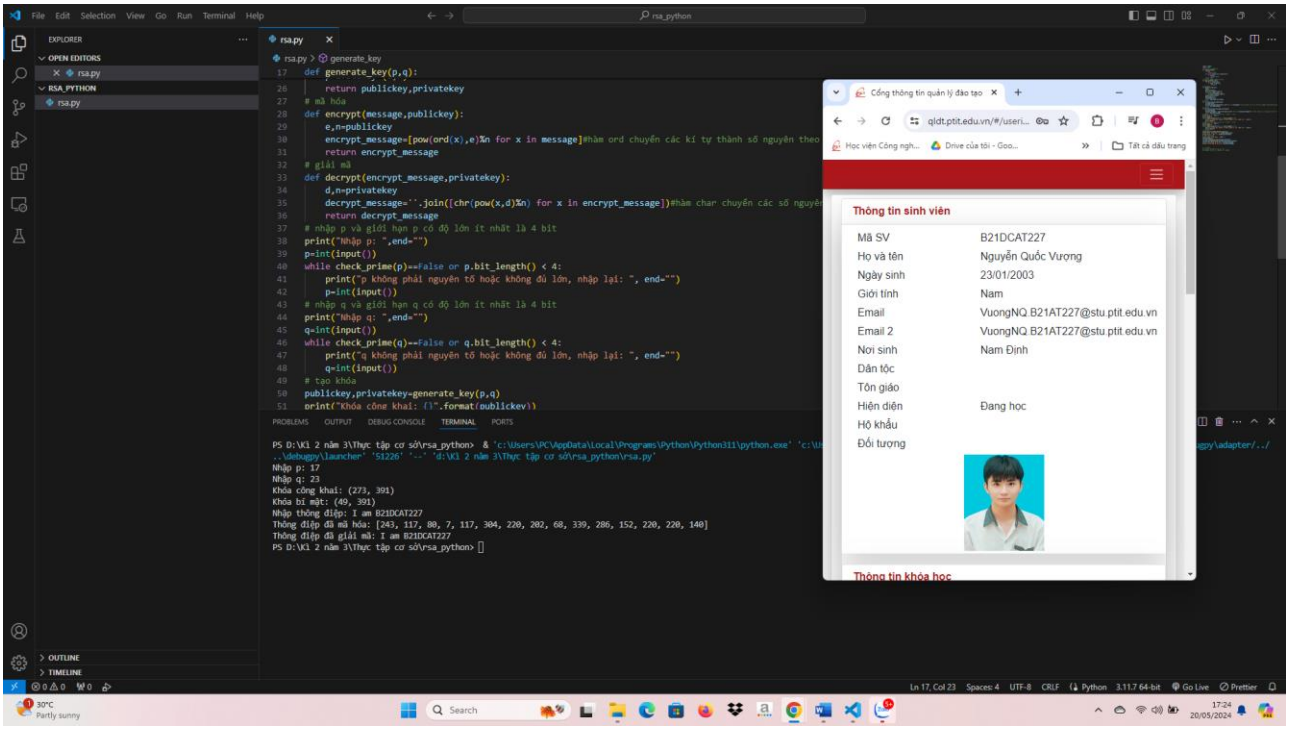


- Lập trình giải thuật mã hóa và giải mã

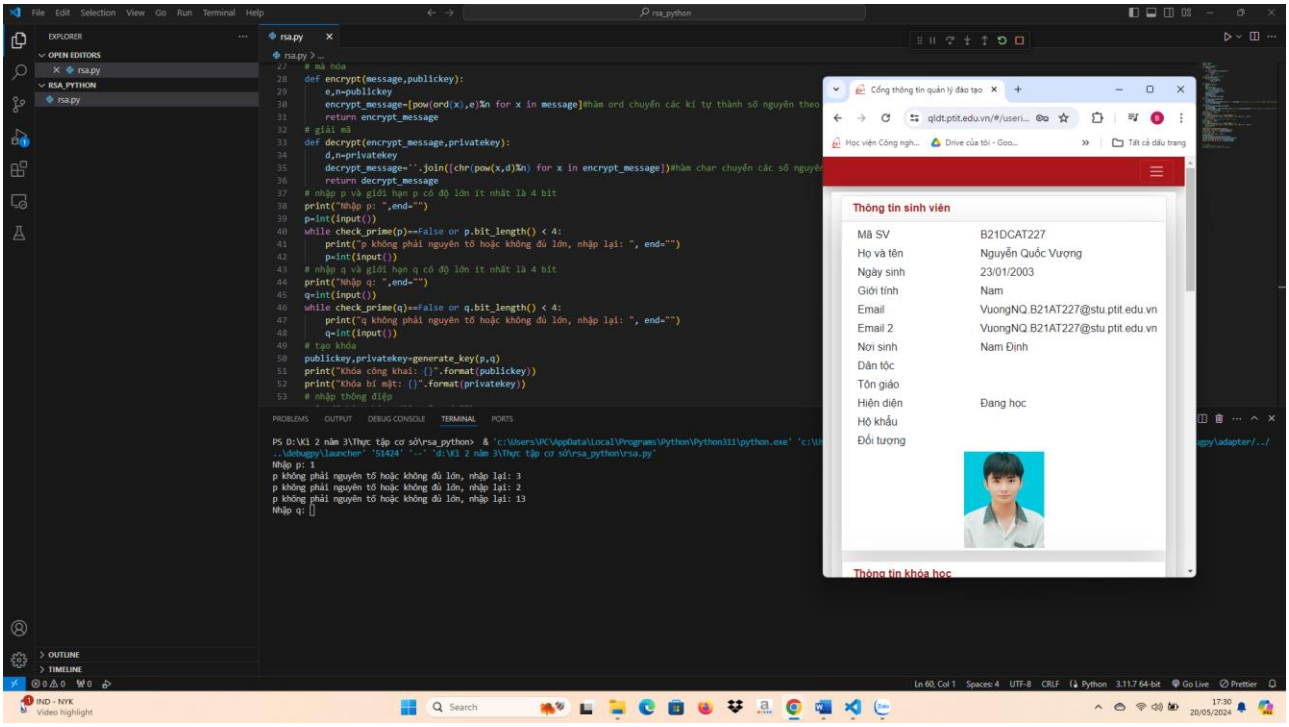




- Thử nghiệm mã hóa và giải mã chuỗi ký tự: “I am <mã sinh viên>” (thay bằng mã sinh viên của mình vào)



Trường hợp số đầu vào không thỏa mãn:


rsa.py
# mã hóa
def encrypt(message,publickey):
e,n=publickey
encrypt_message=[pow(ord(x),e)%n for x in message]#hàm ord chuyển các kí tự thành số nguyên theo
return encrypt_message
# giải mã
def decrypt(encrypt_message,privatekey):
d,n=privatekey
decrypt_message=''.join([chr(pow(x,d)%n) for x in encrypt_message])#hàm char chuyển các số nguyê
return decrypt_message
# nhập p và giới hạn p có độ lớn ít nhất là 4 bit
print("Nhập p: ",end="")
p=int(input())
while check_prime(p)==False or p.bit_length() < 4:
print("p không phải nguyên tố hoặc không đủ lớn, nhập lại: ", end="")
p=int(input())
# nhập q và giới hạn q có độ lớn ít nhất là 4 bit
print("Nhập q: ",end="")
q=int(input())
while check_prime(q)==False or q.bit_length() < 4:
print("q không phải nguyên tố hoặc không đủ lớn, nhập lại: ", end="")
q=int(input())
# tạo khóa
publickey,privatekey=generate_key(p,q)
print("Khóa công khai: {}".format(publickey))
print("Khóa bí mật: {}".format(privatekey))
# nhập thông điệp
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\K1 2 năm 3\Thực tập cơ sở\rsa_python> & 'c:\Users\PC\AppData\Local\Programs\Python\Python311\python.exe' 'c:\U
..\debugpy\launcher' '51424' '--' 'd:\K1 2 năm 3\Thực tập cơ sở\rsa_python\rsa.py'
Nhập p: 1
p không phải nguyên tố hoặc không đủ lớn, nhập lại: 3
p không phải nguyên tố hoặc không đủ lớn, nhập lại: 2
p không phải nguyên tố hoặc không đủ lớn, nhập lại: 13
Nhập q:
Cổng thông tin quản lý đào tạo
qldt.ptit.edu.vn/#/useri...
Thông tin sinh viên
Mã SV B21DCAT227
Họ và tên Nguyễn Quốc Vượng
Ngày sinh 23/01/2003
Giới tính Nam
Email VuongNQ.B21AT227@stu.ptit.edu.vn
Email 2 VuongNQ.B21AT227@stu.ptit.edu.vn
Nơi sinh Nam Định
Dân tộc
Tôn giáo
Hiện diện Đang học
Hộ khẩu
Đối tượng
Thông tin khóa học