|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  KHOA CÔNG NGHỆ THÔNG TIN  ---------------o0o---------------        **BÀI TẬP LỚN**  **MÔN HỌC: AN TOÀN BẢO MẬT THÔNG TIN**   |  | | --- | |  |   Giảng viên hướng dẫn: Đỗ Văn Đức  Sinh viên thực hiện: Ngô Văn Khải  Mã sinh viên: 191200552  Lớp: Công nghệ thông tin 2 – K60  **Hà Nội, tháng 05 năm 2022** |

**Mục Lục**

[**Phần I: Mã hóa cổ điển** 3](#_Toc102600255)

[**1) Mật mã Ceasar** 3](#_Toc102600256)

[**2) Mật mã Hill** 3](#_Toc102600257)

[**3) Mật mã hoán vị** 5](#_Toc102600258)

[**4) Mật mã đơn bảng chữ - Monoalphabet** 5](#_Toc102600259)

[**5) Mật mã Playfair** 6](#_Toc102600260)

[**6) Mật mã Vigenere** 7](#_Toc102600261)

[**7) Mật mã Vigenere – AutoKey** 7](#_Toc102600262)

[**Phần II: Mã hóa hiện đại** 9](#_Toc102600263)

[**1) Mật mã AES (Advanced Encryption Standard)** 9](#_Toc102600264)

[**2) Mã hóa DES (Data Encryption Standard)** 13](#_Toc102600265)

[**Phần III: Modulo** 18](#_Toc102600266)

[**1. Tính lũy thừa modulo bằng hạ bậc** 18](#_Toc102600267)

[**2. Nghịch đảo modulo – thuật toán euclid mở rộng** 18](#_Toc102600268)

[**3. Tính lũy thừa modulo sử dụng định lý fermat** 19](#_Toc102600269)

[**4. Tính lũy thừa modulo bằng định lý euler** 20](#_Toc102600270)

[**5. Căn nguyên thủy** 21](#_Toc102600271)

[**6. Logarit rời rạc**  22](#_Toc102600272)

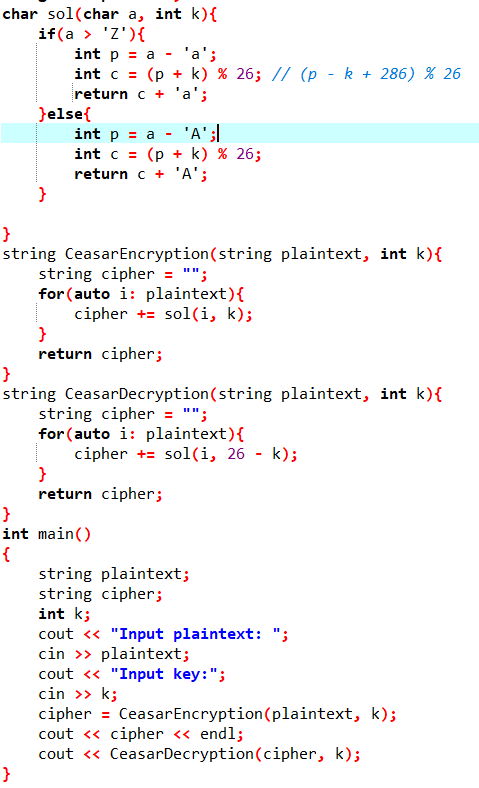
# **Phần I: Mã hóa cổ điển**

## **1) Mật mã Ceasar**

a) Ý tưởng:

- Mã hóa: Thay thế các chữ cái từ chuỗi plaintext bằng các ký tự ở bảng chữ cái đã mã hóa. Bảng chữ cái đã mã hóa là bảng chữ cái tiếng anh đã được dịch sang phải k lần. Biểu diễn toán học: **(p + k) mod 26**

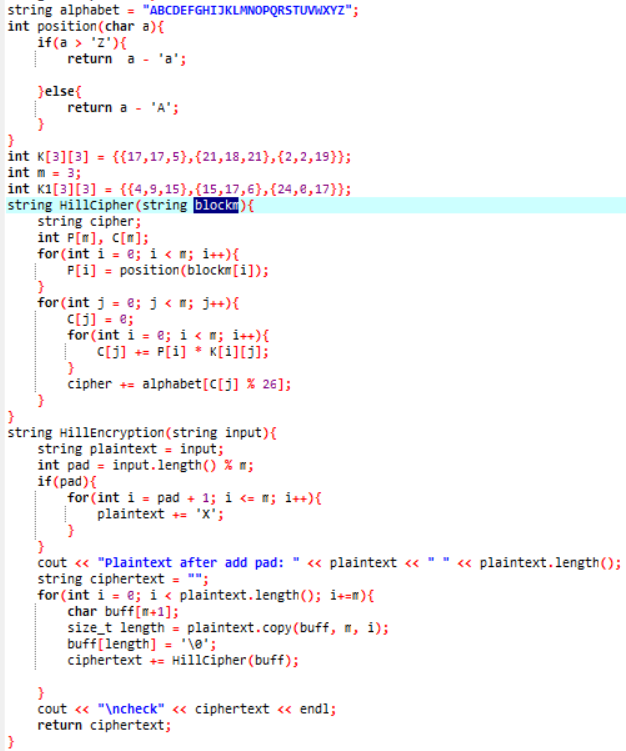
- Giải mã: Đảo ngược các bước ở trên

b) Code

## **2) Mật mã Hill**

a) Ý tưởng:

- Lẩy m tổ hợp tuyến tính của m ký tự chữ cái trong một phần tử của văn bản gốc, theo đó tính ra được m ký tự chữ cái trong một phần tử văn bản mã hóa

b) Code

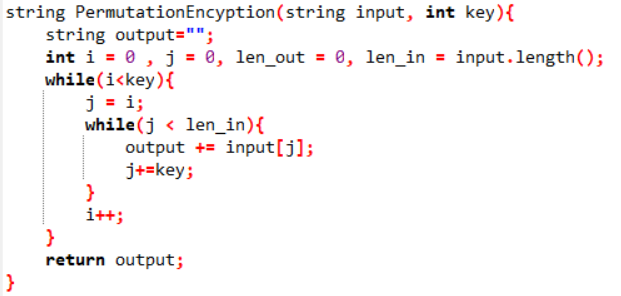
## **3) Mật mã hoán vị**

a) Ý tưởng

- Tách bản rõ thành n hàng dọc theo khóa đầu vào

- Bản mã là được ghép từ các hàng vừa viết

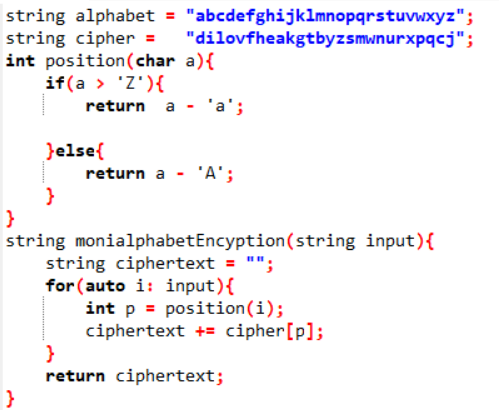
b) Code



## **4) Mật mã đơn bảng chữ - Monoalphabet**

a) Ý tưởng

- Mỗi ký tự của bản rõ được thay thế bằng ký tự khác trong bản mã (có thể là một chữ cái, môṭ số hoăc̣ môṭ ký hiêụ).  
- Khóa của hệ mã chính là thứ tự các ký tự được thay thế tương ứng theo một key cho trước

b) Code

## **5) Mật mã Playfair**

a) Ý tưởng

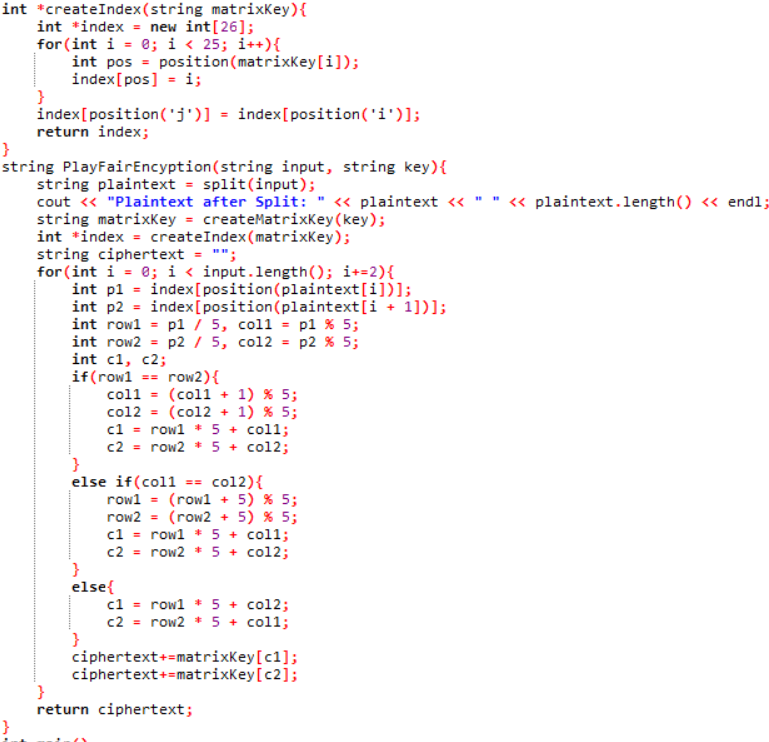
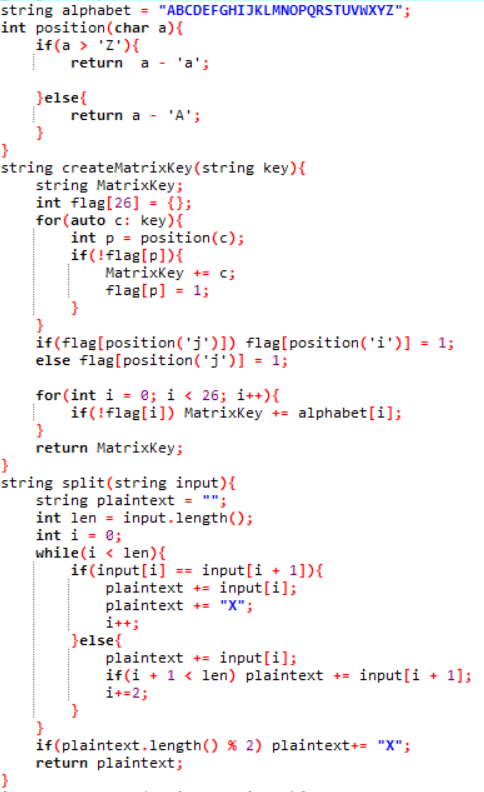
- Tách các ký tự trong bảng gốc thành tập hợp gồm 2 ký tự liên tiếp nhau. Nếu hai ký tự liên tiếp nhau giống nhau thì ta tách thành 2 ký tự và thêm x vào ký tự xuất hiện trước

- Xây dựng ma trận khóa từ khóa cho trước và thêm các ký tự còn thiếu trong bảng chữ cái. Lưu ý i và j là 1.

- Hai ký tự thuộc một hàng thì thay thế bằng 2 ký tự tương ứng bên phải (có nối vòng)

- Hai ký tự thuộc một cột thì thay thế bằng 2 ký tự bên dưới (có nối vòng)

- Hai ký tự lập thành hình chữ nhật thì thay thế bằng 2 ký tự tương ứng trên cùng dòng ở góc còn lại của hình chữ nhật.

b) Code

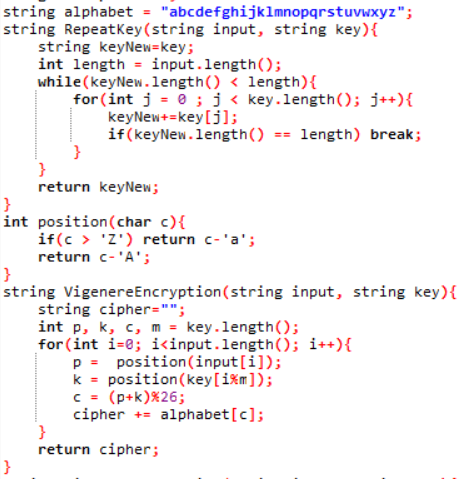
## **6) Mật mã Vigenere**

a) Ý tưởng

- Lặp khóa đến khi độ dài của khóa bằng với độ dài của bản gốc.

- Tra bảng theo từng cặp ký tự ở khóa và bản gốc

b) Code



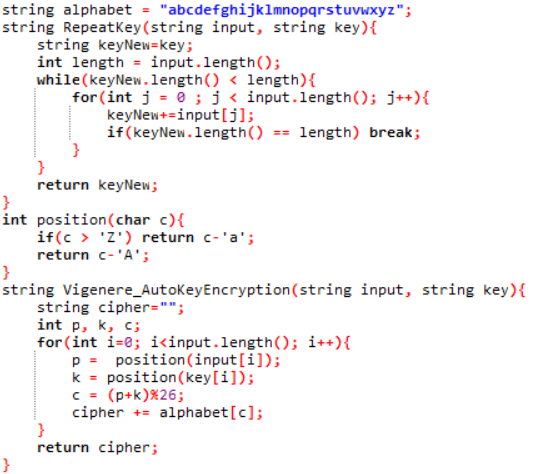
## **7) Mật mã Vigenere – AutoKey**

a) Ý tưởng:

- Tạo khóa bằng cách ghép khóa cho trước và thêm các ký tự của bản gốc sao cho độ dài khóa bằng với độ dài bản gốc

- Tra bảng theo từng cặp ký tự ở khóa và bản gốc

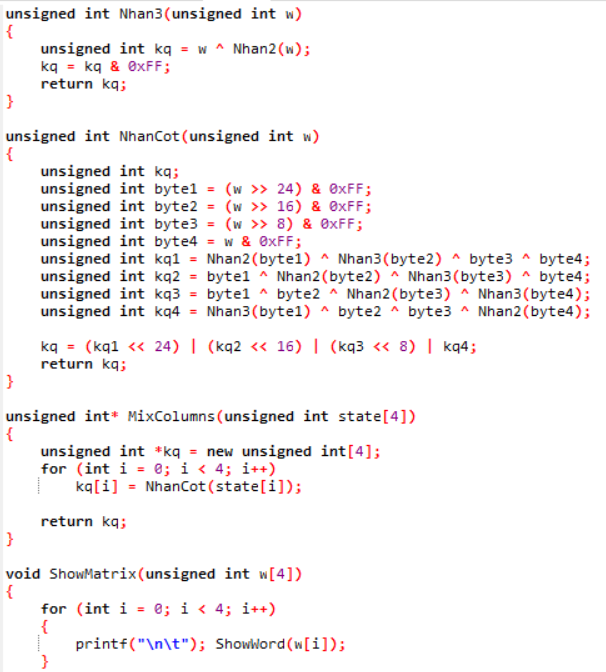
b) Code

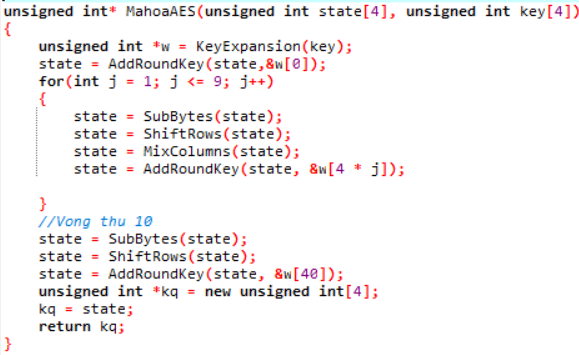


# **Phần II: Mã hóa hiện đại**

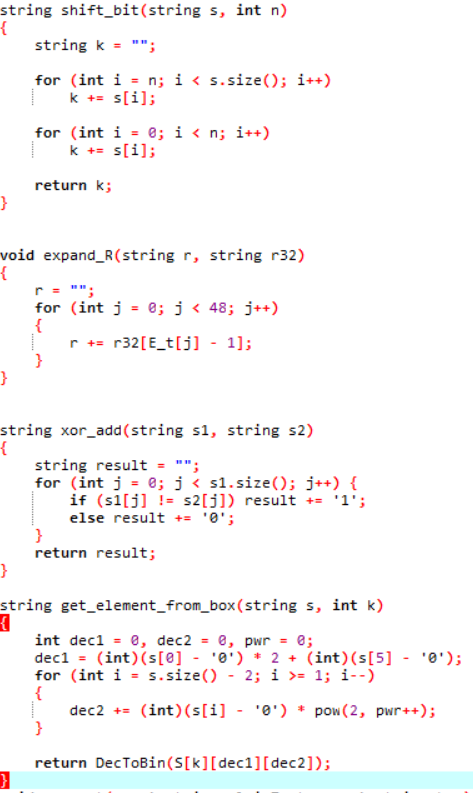
## **1) Mật mã AES (Advanced Encryption Standard)**

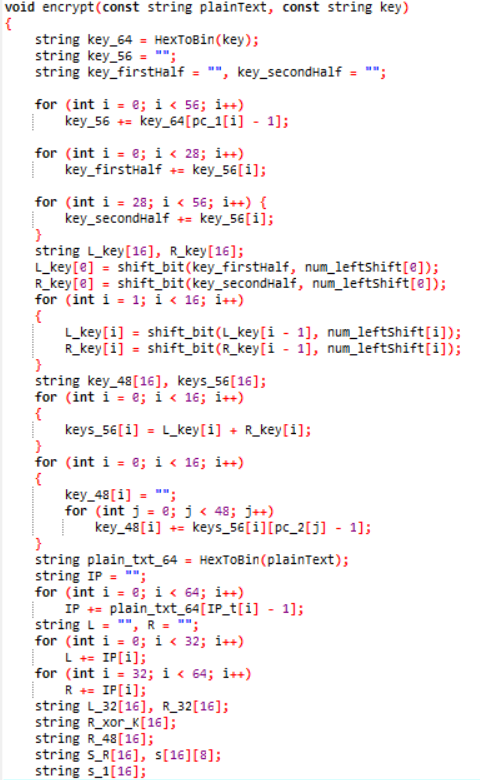


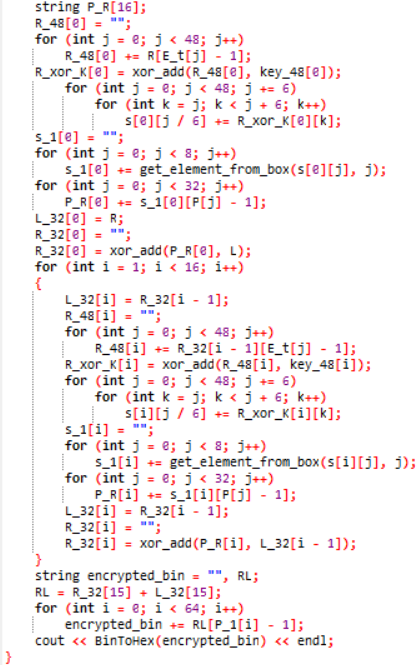


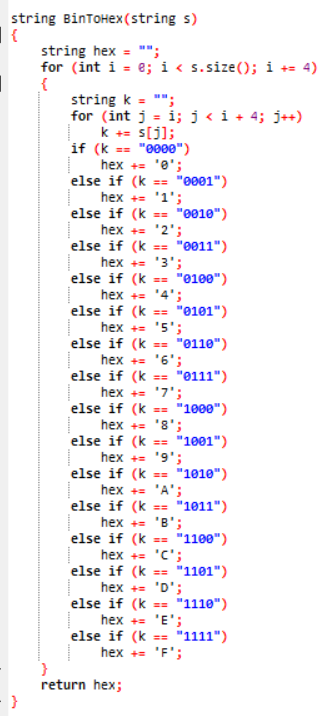


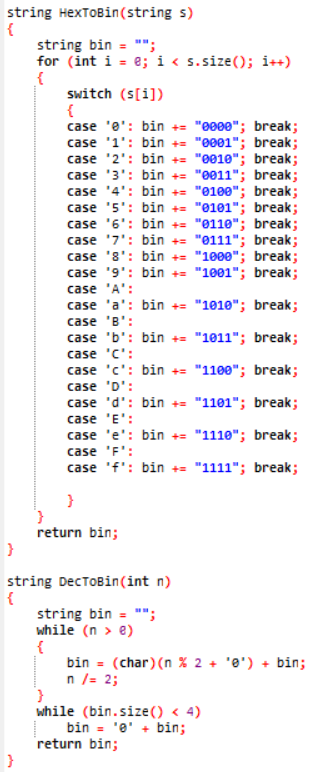
## **2) Mã hóa DES (Data Encryption Standard)**











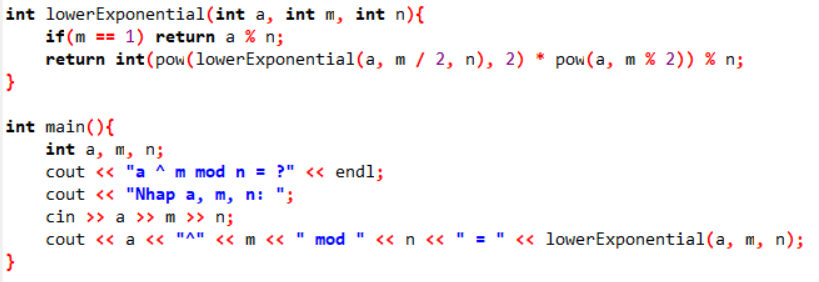
# **Phần III: Modulo**

## **1. Tính lũy thừa modulo bằng hạ bậc**

a) Ý tưởng:

- Dùng đệ quy để hạ bậc dần bậc xuống mức ta có thể tính được kết quả.

b) Code

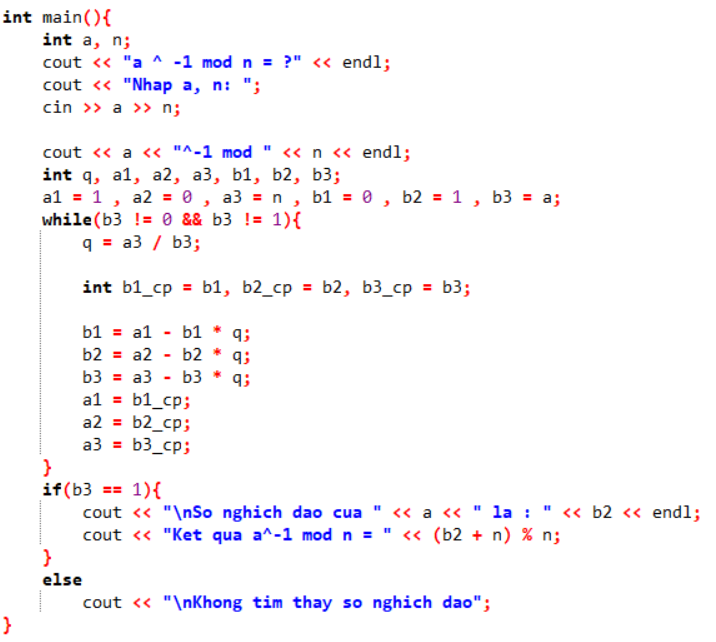


## **2. Nghịch đảo modulo – thuật toán euclid mở rộng**

a) Ý tưởng: Euclid mở rộng luôn có dạng ri = Axi + Byi

- Áp dụng công thức sau: Lặp lại cho đến khi ri = 1 hoặc ri = 0.

b) Code



## **3. Tính lũy thừa modulo sử dụng định lý fermat**

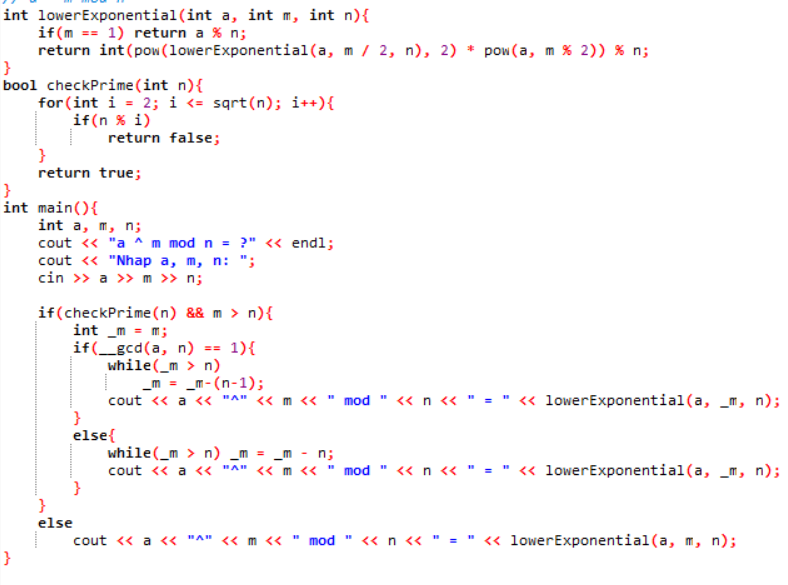
a) Ý tưởng:

- Kiểm tra xem a có phải số nguyên dương và chia hết cho n hay không

+ Nếu a là số nguyên dương và không chia hết cho n thì an – 1 (mod n)

+ Nếu a là số nguyên dương thì an

b) Code:



## **4. Tính lũy thừa modulo bằng định lý euler**

a) Ý tưởng:

- Tính

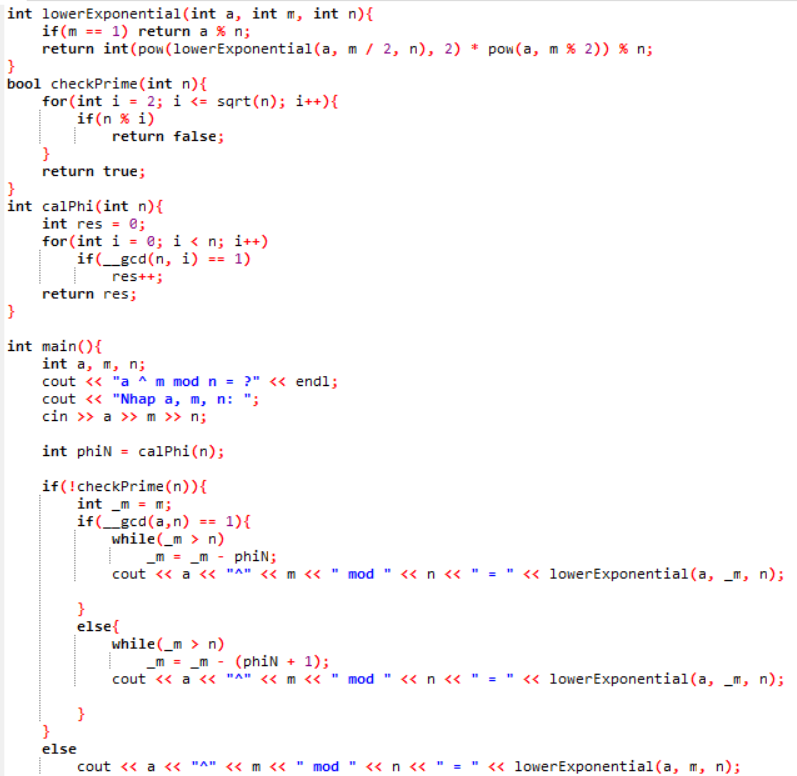
- Kiểm tra xem số chia có phải là số nguyên tố không.

+ Nếu phải thì là dạng 1: hạ bậc mũ về tích của một số với và cộng phần dư. => Kết quả của bài toán được tính thông qua cách hạ bậc với mũ mới là mũ sau khi hạ.

+ Nếu không thì là dạng 2: hạ bậc mũ về tích của một số với ( + 1) và cộng phần dư. => Kết quả của bài toán được tính thông qua cách hạ bậc với mũ mới là mũ sau khi hạ.

- Nếu số chia không phải số nguyên tố thì hạ trực tiếp.

b) Code



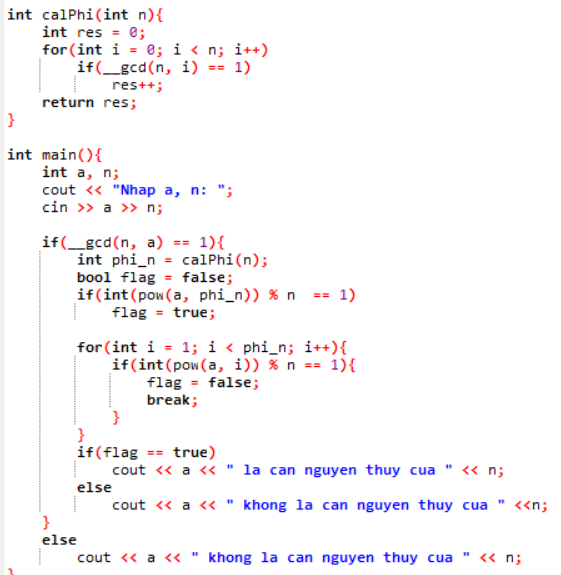
## **5. Căn nguyên thủy**

a) Ý tưởng:

- Tính .

- Kiểm tra a và các số nhỏ hơn phi-n có là nguyên tố cùng nhau không. Nếu một cái không là nguyên tố cùng nhau thì không phải căn nguyên thủy. Ngược lại là căn nguyên thủy

b) Code



## **6. Logarit rời rạc** a) Ý tưởng:

- Kiểm tra a có phải là căn nguyên thủy của n hay không. Nếu không thì không có logarit rời rạc.

- Nếu có ta tìm xem ai mod n có bằng b hay không (i chạy từ 1 đến ). Nếu tồn tại ai mod n = b thì i là kết quả cần tìm. Ngược lại thì không có logarit rời rạc

b) Code

