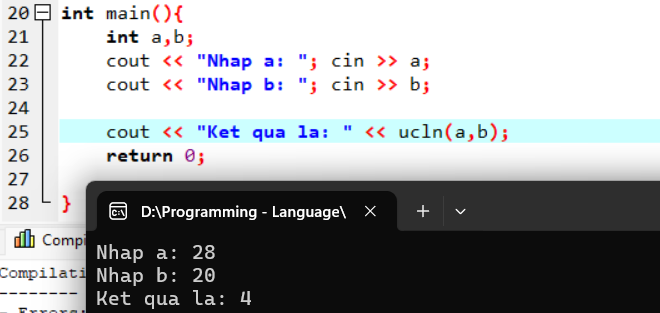
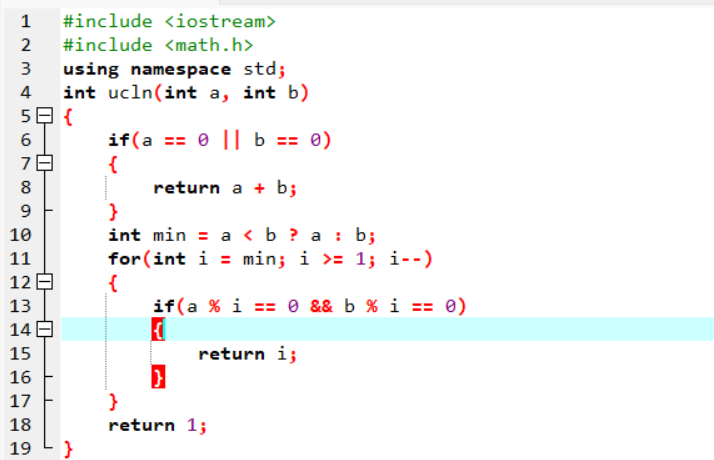
**1. Thuật Toán Ngây Thơ**

Ước Chung Lớn Nhất (UCLN) của 2 số a, b là số lớn nhất mà 2 số a và b cùng chia hết

Thuật toán tự nhiên mà bạn dùng để tìm UCLN đó là duyệt từ số nhỏ hơn trong 2 số về tới 1, số nào mà cả a và b chia hết đầu tiên sẽ là UCLN

Chú ý rằng UCLN(a, 0) = a

Code 1 :



Bội Chung Nhỏ Nhất (BCNN) của 2 số a, b là số nhỏ nhất mà chia hết đồng thời cho cả a và b

Thuật toán tự nhiên mà bạn dùng để tìm BCNN đó là duyệt từ số lớn hơn trong 2 số và tăng dần cho tới khi gặp số đầu tiên chia hết cho cả a và b.

**2. Thuật Toán Euclid**

Thuật toán Euclid phát biểu như sau :  UCLN của hai số nguyên không thay đổi khi thay số lớn hơn bằng hiệu của nó với số nhỏ hơn

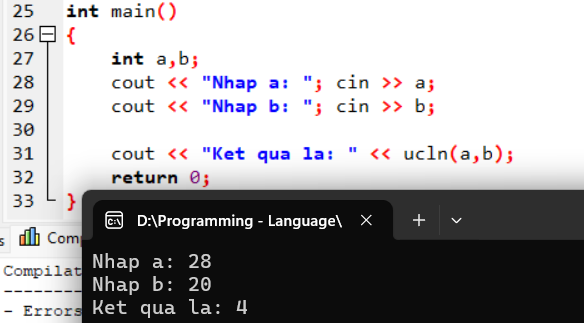
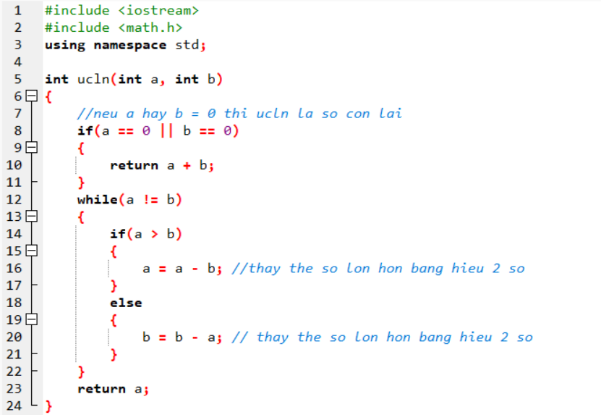
Quá trình thay thế này được lặp đi lặp lại cho tới khi 2 số bằng nhau, khi đó UCLN chính là 1 trong 2 số.

Ví dụ

UCLN(20, 15) = UCLN (15, 5) = UCLN(5, 10) = UCLN(5, 5) => UCLN(20, 15) = 5

UCLN(16, 10) = UCLN(10, 6) = UCLN(6, 4) = UCLN(4, 2) = UCLN(2, 2) => UCLN(16, 10) = 2

Code 2:

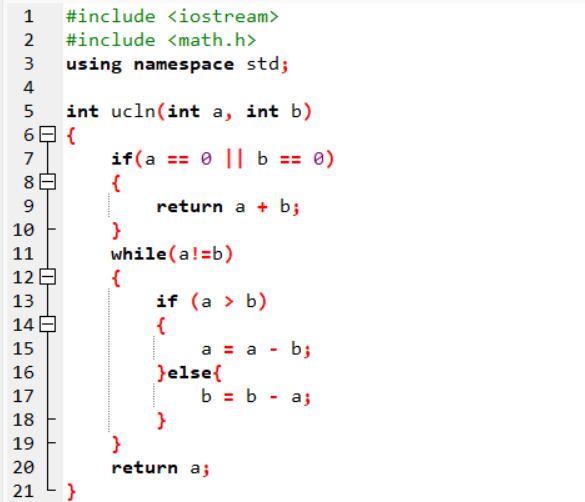


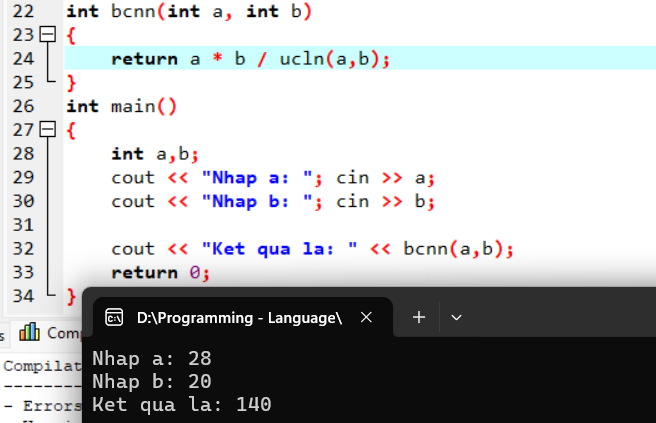
Giải thích :

1. a = 28, b = 20, a != b vòng lặp lặp lần 1 : vì a > b nên a = a - b = 28 - 20 = 8
2. a = 8, b = 20, a != b vòng lặp lặp lần 2 : vì b > a nên b = b - a = 20 - 8 = 12
3. a = 8, b = 12, a != b vòng lặp lặp lần 3 : vì b > a nên b = b - a = 12 - 8 = 4
4. a = 8, b = 4, a != b vòng lặp lặp lần 3 : vì a > b nên a = a - b = 8 - 4 = 4
5. a = 4, b = 4, a == b => Vòng lặp kết thúc và hàm trả về 4

Để tìm BCNN bạn có thể dựa vào kiến thức : BCNN(a, b) = a \* b / UCLN(a, b).

Ví dụ BCNN(28, 20) = 28 \* 20 / UCLN(28, 20) = 560 / 4 = 140

Code 3 :



**3. Cải Tiến Thuật Toán Euclid**

Trong mục 2 khi a và b cách xa nhau thì thuật toán Euclid hoạt động không hiệu quả, ví dụ bạn tìm ULCN(1000000000, 1) thì thuật toán cần lặp 999999999 lần.

Ta cải tiến thuật toán Euclid bằng nhận xét sau :  UCLN của hai số nguyên không thay đổi khi thay 1 trong 2 số thành số dư của nó với số còn lại. Có nghĩa là  *UCLN(a, b) = UCLN(b, a % b)*

Quá trình thay thế này được lặp đi lặp lại cho tới khi 1 trong 2 số bằng 0, khi đó UCLN chính là số còn lại

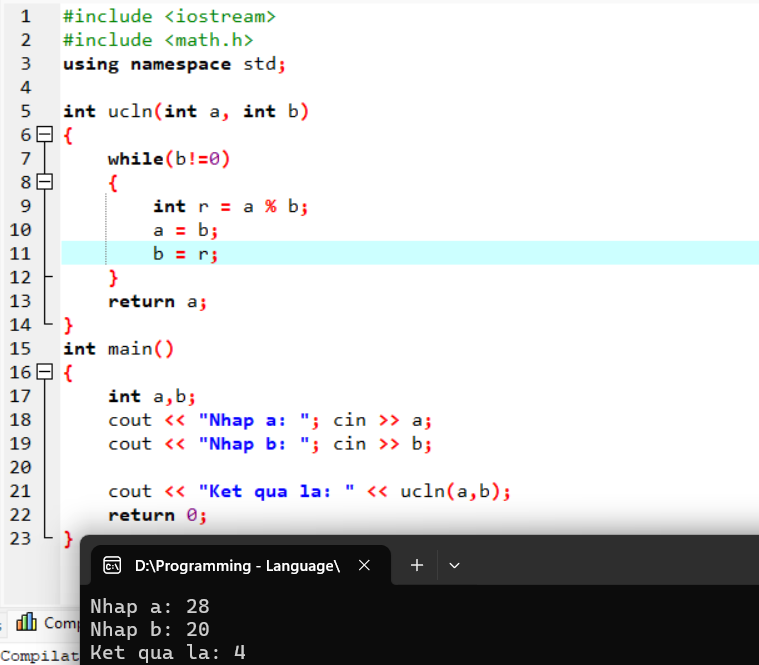
Ví dụ

UCLN(20, 15) = UCLN(15, 20 % 15) = UCLN(15, 5) = UCLN(5, 15 % 5) = UCLN(5, 0) => UCLN(20, 15) = 5

UCLN(16, 28) = UCLN(28, 16 % 28) = UCLN(28, 16) = UCLN(16, 28 % 16) = UCLN(16, 12) = UCLN(12, 16 % 12) = UCLN(12, 4) = UCLN(4, 12 % 4) = UCLN(4, 0) => UCLN(16, 28) = 4

UCLN(1000000000, 1) = UCLN(1, 1000000000 % 1) = UCLN(1, 0) = 1 => UCLN(1000000000, 1) = 1 (Bạn chỉ cần 1 vòng lặp thay vì 999999999)

Code 4:



Giải thích :

1. a = 28, b = 20, b != 0 nên vòng lặp lặp lần 1 : r = a % b = 8, a = b = 20, b = r = 8

2. a = 20, b = 8, b != 0 nên vòng lặp lặp lần 2 : r = a % b = 4, a = b = 8, b = r = 4

3. a = 8, b = 4, b != 0 nên vòng lặp lặp lần 3 : r = a % b = 0, a = b = 4, b = r = 0

4. a = 4, b = 0, b == 0 nên vòng lặp dừng, hàm trả về a = 4

Cải tiến này giúp bạn có thể code hàm tìm UCLN nhanh hơn rất nhiều so với thuật toán Euclid ban đầu từ đó thuật toán tìm BCNN cũng hiệu quả hơn.