

### Cấu trúc lặp (mỗi bài viết bằng 3 loại vòng lặp: for / while / do...while)

- Viết chương trình nhập số nguyên dương n. Liệt kê n số nguyên tố đầu tiên.
- Viết chương trình nhập vào hai số nguyên dương a và b. Tìm ước số chung lớn nhất và bội số chung nhỏ nhất của a và b.

```

if (a == 0 || b == 0){
    return a + b;
}
while (a != b){
    if (a > b){
        a -= b; // a = a - b
    }else{
        b -= a;
    }
}

```

3. *return a; // return a or b, bởi vì lúc này a và b bằng nhau*  
 BSCNN (a \* b) / USCLN(a, b);

- Viết chương trình đếm số ước số của số nguyên dương N.

Ví dụ: N=12

số ước số của 12 là 6

- Một số hoàn thiện là một số có tổng các ước số của nó (không kể nó) bằng chính nó. Hãy liệt kê các số hoàn thiện nhỏ hơn 5000.

Ví dụ: số 6 là số hoàn thiện vì tổng các ước số là  $1+2+3 = 6$ .

- Tính  $n!$ , với  $n \geq 0$
- Tính  $P(n) = 1.3.5...(2n+1)$ , với  $n \geq 0$
- Tính  $S(n) = 1 + 3 + 5 + \dots + (2 \times n + 1)$ , với  $n \geq 0$
- Tính  $S(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1}n$ , với  $n > 0$
- Tính  $S(n) = 1 + 1.2 + 1.2.3 + \dots + 1.2.3...n$ , với  $n > 0$
- Tính  $S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$ , với  $n > 0$
- Tính  $S(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$ , với  $n > 0$
- Tính  $S(n) = 1 + 2^2 + 3^3 + \dots + n^n$ , với  $n \geq 0$
- Tính  $S(n) = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n}{n+1}$ , với  $n > 0$
- Tính  $S(n) = 1 + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$ , với  $n > 0$