

# Chapter 4. Functions and recursive

## Khái niệm chương trình con

- Để thuận tiện cho việc viết chương trình người ta thường tách chương trình đó thành các đoạn chương trình nhỏ; mỗi đoạn chương trình nhỏ này giải quyết *trọn vẹn* một vấn đề/công đoạn cụ thể của bài toán; mỗi đoạn chương trình nhỏ đó được gọi là một **chương trình con**.
- Chương trình con trong C/C++ được gọi chung là **hàm**. Hàm trong C/C++ có hai dạng là hàm có giá trị trả về và hàm không có giá trị trả về.

## Lợi ích của việc sử dụng chương trình con

- Làm cho việc phân tích thiết kế chương trình được thuận lợi.
- Giúp thuận lợi trong việc kiểm thử, nâng cấp, viết tài liệu cho chương trình, rút ngắn thời gian lập trình, tránh lặp lại các đoạn lệnh tương tự nhau,...
- Bản thân ngôn ngữ lập trình C/C++ đã được hỗ trợ một số hàm; các hàm này được gọi là các hàm chuẩn.
- Tùy theo nhu cầu thực tế mà người lập trình có thể thiết kế thêm các hàm khác; các hàm này được gọi là các hàm tự tạo.

## Các loại tham số

- Các biến liệt kê trong các dòng *tiêu đề hàm* thường được gọi là các **tham số**; các tham số này gọi chung là tham số **hình thức**. Hàm có thể không có tham số hoặc có nhiều tham số.
- Các tham số mà ngay trước nó không có chỉ dẫn & thì được hiểu là tham số đó được truyền theo kiểu **tham trị**,
- Các biến mà ngay trước nó có chỉ dẫn & thì được hiểu là tham số đó được truyền theo kiểu **tham biến**.
- Danh sách các tham số ở *lời gọi hàm* được gọi là **tham số thực tế**.

# Phân biệt tham số hình thức trị/biến

- Các tham số được truyền theo kiểu tham trị thì những thay đổi trong thân hàm chứa nó sẽ không được giữ lại khi ra khỏi hàm chứa nó;
- Ngược lại, nếu tham số được truyền theo kiểu tham biến thì mọi sự thay đổi trong thân hàm đó sẽ được giữ lại khi ra khỏi hàm đó.
- Tham số được truyền theo kiểu tham biến là phương thức quan trọng để thiết kế hàm trả về nhiều giá trị.

## Biến toàn cục và biến địa phương

- Các biến được khai báo trong các hàm - kể cả hàm `int main ()` được gọi là các **biến cục bộ** (biến địa phương) và nó chỉ có ảnh hưởng trong phạm vi thân hàm chứa nó. Khi hàm chứa nó kết thúc thì các biến này cũng mất tác dụng theo.
- Ngược lại, các biến được khai báo ngoài được gọi là các **biến toàn cục**; biến toàn cục ảnh hưởng đến cả chương trình.
- Khi lập trình chúng ta cần tránh tối đa việc sử dụng các biến toàn cục.

## Hàm không có giá trị trả về, hàm có giá trị trả về

- Hàm không có giá trị trả về (trả về kiểu *void*), hàm có giá trị trả về (trả về kiểu khác *void*).
- Nếu hàm cần thiết kể chỉ có một giá trị trả về thì có thể thiết kế hàm theo dạng hàm không có giá trị trả hoặc có giá trị trả về; khi đó ta cần thêm thông tin sau để quyết định việc lựa chọn một trong hai cách trên:
  - Nếu tên hàm không được sử dụng trong các hàm hoặc biểu thức khác nữa thì nên thiết kế theo dạng **hàm không có giá trị trả về**,
  - ngược lại thì nên thiết kế theo dạng **hàm có giá trị trả về**.
  - Hàm được thiết kế theo dạng có giá trị trả về thì trong thân hàm đó phải có lệnh `return` để trả về giá trị là kết quả cần trả về của hàm đó.

Khi cần thiết kể một hàm có nhiều hơn một giá trị thì có thể giải quyết theo ba cách sau:

- *thứ nhất*, phân rã chức năng của hàm thành nhiều chức năng nhỏ hơn nữa, để mỗi hàm có đúng một chức năng riêng biệt và nó sẽ trả về đúng một giá trị;
- *thứ hai*, trả về nhiều giá trị thông qua kỹ thuật truyền tham biến – cần trả về bao nhiêu giá trị thì sử dụng thêm bấy nhiêu tham biến;
- *thứ ba*, sử dụng biến toàn cục – dùng biến toàn cục để ghi nhận giá trị cần trả về.



# Ví dụ 1

Viết chương trình nhập vào hai phân số  $a/b$  và  $c/d$ .

Hãy tính tổng, hiệu, tích, thương của hai phân số.

Yêu cầu phân số kết quả phải ở dạng tối giản.

Ví dụ1:  $1/2 + 1/6$  có kết quả là  $2/3$ .

Ví dụ 2:  $1/2 + 3/4$  có kết quả là  $5/4$ .

# HÀM ĐỆ QUY

- Một hàm được gọi là có tính đệ quy nếu trong bản thân hàm đó có lệnh gọi lại chính nó một cách trực tiếp hay gián tiếp.
- Một chương trình được gọi là có đệ quy nếu nó có chứa ít nhất một hàm đệ quy (gọi tắt là chương trình đệ quy).

# Sơ đồ hàm đệ quy

Một hàm đệ quy (recursive function) gồm 2 bước:

- **Bước cơ sở** (hay còn gọi là bước dừng):

Mô tả cấp độ giải được của bài toán.

- **Bước đệ quy:**

Bước gọi lại chính nó nhưng với cấp độ thấp hơn.

# Hàm đệ qui hoạt động như thế nào ?

- Khi một hàm gọi đệ quy đến chính nó, chương trình sẽ tạo ra một tập các biến cục bộ hoàn toàn độc lập với các tập biến cục bộ đã được tạo ra trong các lần gọi trước đó.
- Có bao nhiêu lần gọi tới hàm đệ quy thì cũng có bấy nhiêu lần thoát ra khỏi hàm.
- Cứ mỗi lần thoát ra khỏi hàm thì một tập các biến cục bộ sẽ được giải phóng.
- Sự tương ứng giữa các lần gọi tới hàm và thoát ra khỏi hàm được thực hiện theo thứ tự ngược, nghĩa là lần ra đầu tiên ứng với lần vào cuối cùng và lần ra khỏi hàm cuối cùng ứng với lần đầu tiên gọi tới hàm (cơ chế vào sau ra trước).

# PHÂN LOẠI ĐỆ QUY

- Đệ quy tuyến tính,
  - Đệ quy nhị phân,
  - Đệ quy hồi tưởng,
  - Đệ quy phi tuyến
- ❖ (việc phân loại này chỉ mang tính hình thức)

# Đệ quy tuyến tính

Một hàm được gọi là đệ quy tuyến tính (đệ quy đơn) nếu một lần gọi hàm nó chỉ phát sinh tối đa một lời gọi đệ quy.

## Ví dụ 2.

Giai thừa được định nghĩa theo kiểu quy nạp như sau:  $n! = n \cdot (n-1)!$   
Viết chương trình tính  $n!$ , với  $n$  là một số nguyên không âm.

```
1. long giaithua(int n)
2. {
3.     if (n==0 || n==1)
4.         return 1;
5.     else
6.         return n*giaithua(n-1);
7. }
```

# Đệ quy nhị phân

Một hàm được gọi là đệ quy nhị phân nếu mỗi lần gọi hàm nó phát sinh một số ít lời gọi đệ quy.



### Ví dụ 3.

Viết chương trình tính số hạng thứ  $n$  của dãy fibonacci  $f_n$  xác định theo công thức đệ quy sau:

$$f_1=1,$$

$$f_2=1,$$

$$f_n = f_{n-1} + f_{n-2}, n \geq 3.$$

1. *int f(int n)*

2. {

3.     *if (n==1 || n==2) return 1;*

4.     *else return f(n-1)+f(n-2);*

}

## Ví dụ 4.

*Cho một dãy số được định nghĩa theo công thức quy nạp sau (với  $n$  là số nguyên  $\geq 1$ )*

$$f(1) = 1; f(2) = 2; f(3) = 3.$$

$$f(n + 3) = 2f(n + 2) + f(n + 1) - 3f(n)$$

*Viết chương trình tính  $f(n)$ .*

1. long f(int n)

2. {

3.     if (n==1 || n==2 || n==3) return n;

4.     return 2\*f(n-1) + f(n-2)-3\*f(n-3);

}

# Đệ quy hỗ tương

Hai hàm P,Q được gọi là đệ quy hỗ tương nếu hàm P có lời gọi đến hàm Q và ngược lại một cách trực tiếp hay gián tiếp.

Ví dụ 5.

*Viết chương trình tính số hạng thứ n của hai dãy sau:*

$$x_0 = 1, y_0 = 0,$$

$$x_n = x_{n-1} + y_{n-1} \text{ với mọi } n > 0,$$

$$y_n = 3 * x_{n-1} + 2 * y_{n-1} \text{ với mọi } n > 0.$$

1. *int tinhxn(int n)*

2. {

3.     *if (n==0) return 1;*

4.     *return tinhxn(n-1)+tinhhyn(n-1);*

5. }

6. *int tinhhyn(int n)*

7. {

8.     *if (n==0) return 0;*

9.     *return 3\*tinxn(n-1)+2\*tinhyn(n-1);*

}

# Đệ quy phi tuyến

- Một hàm được gọi là đệ quy phi tuyến (đệ quy phức) nếu mỗi lần gọi hàm thì nó phát sinh ra khoảng  $n$  lần gọi đệ quy.
- Thông thường với loại này, lời gọi đệ quy được đặt trong một hoặc nhiều vòng lặp.

## Ví dụ 6.

Dãy  $A_n$  được cho như sau :

$$A_1=1$$

$$A_n=n(A_1+A_2+\dots+A_{n-1})$$

Viết hàm tính  $A_n$  có sử dụng đệ quy

```
1. int An(int n)
2. {
3.     if (n==1) return 1 ;
4.     int s=0 ;
5.     for (int i=1;i<n ;i++)
6.         s=s+An(i) ;
7.     return s*n ;
8. }
```

# KHỬ ĐỆ QUY

- Không nên lạm dụng đệ quy và nếu một bài toán không quá khó để tìm được một lời giải không đệ quy thì nên chọn cách giải không đệ quy.
- Lời giải không đệ quy được gọi là khử đệ quy.
- SV dễ dàng khử đệ quy cho các bài toán tính  $n!$ , tính số hạng thứ  $n$  của dãy fibonacci

## BT1

Hãy viết **chương trình hoàn chỉnh** nhập một số nguyên dương  $n$  ( $n \leq 10^4$ ) và thực hiện các công việc sau đây.

1. Tìm một số nguyên dương nhỏ hơn hoặc bằng  $n$  mà tổng các chữ số của nó là lớn nhất; xuất tổng lớn nhất tìm được. Đếm xem từ 1 đến  $n$  có bao nhiêu số nguyên tố mà tất cả các chữ số của nó cũng là số nguyên tố ?
2. Đếm xem từ 1 đến  $n$  có bao nhiêu số hoàn chỉnh ? Có bao nhiêu số Armstrong ?
3. Đếm xem từ 10 đến  $n$  có bao nhiêu số nguyên tố kép ? (số nguyên tố kép là số nguyên tố mà khi viết các chữ số của nó theo chiều từ trái qua phải hay từ phải qua trái thì đều được là số nguyên tố; ví dụ số 13, 17, 101, 107).



4.Đếm xem từ 1 đến  $n$  có bao nhiêu số nguyên tố đối xứng ? (số nguyên tố đối xứng là một số nguyên tố và nó bằng trung bình cộng của hai số nguyên tố liền trước và liền sau nó).

5.Tìm cặp số nguyên tố  $(x,y)$  với  $x,y$  là hai số nguyên tố liên tiếp nhỏ hơn hoặc bằng  $n$  sao cho khoảng cách giữa  $x$  và  $y$  là lớn nhất ( $2 \leq x,y \leq n$ ). Xuất khoảng cách lớn nhất đó; khoảng cách bằng  $|x-y|$ .

6.Đếm xem có bao nhiêu cặp số nguyên dương  $x,y$  có tổng bằng  $n$  ? với điều kiện  $x$  khác  $y$ ;  $(x,y)$  và  $(y,x)$  được xem như 1 cặp. Có bao nhiêu bộ 3 số nguyên dương  $x,y,z$  có tổng bằng  $n$  ? với  $x,y,z$  khác nhau đôi một và bộ 3 số  $(x,y,z)$  không kể thứ tự.

7. Phân tích số  $n$  thành tích các thừa số nguyên tố (ví dụ  $90 = 2 \cdot 3 \cdot 3 \cdot 5$ ).
8. Tìm ước số chung lớn nhất của các chữ số của số  $n$ , tìm bội số chung nhỏ nhất của các chữ số của số  $n$  (quy ước:  $\text{UCLN}(x,0)=x$ ;  $\text{BCNN}(x,0)=0$ ; giả sử  $\text{UCLN}$ ,  $\text{BCNN}$  của 2 số tự nhiên  $x,y$  thì  $x,y$  không đồng thời bằng 0). Giả thiết câu này  $n$  có ít nhất 2 chữ số.
9. Các số chính phương viết liên nhau như sau 149162536496481100121.... Hỏi tại vị trí thứ  $n$  của dãy trên là chữ số nào ?
10. Đếm xem có bao nhiêu cặp số  $(p,q)$  sao cho tổng các ước thực sự của  $p$  bằng  $q$  và tổng các ước thực sự của  $q$  bằng  $p$ ; trong đó  $p < q$ ;  $p, q$  nhỏ hơn hoặc bằng  $n$ . Ghi chú: ước thực sự là ước không kể chính nó, ví dụ 6 có 3 ước thực sự là 1,2,3.
11. Hãy tìm 3 số nguyên tố nhỏ nhất lớn hơn 1 tỉ. Hãy tìm 3 số hoàn chỉnh nhỏ nhất lớn hơn 1 tỉ.

Dữ liệu nhập từ bàn phím; kết quả xuất ra màn hình.

Ví dụ:

n=30

Cau 1:	11	5
Cau 2:	2	9
Cau 3:	7	
Cau 4:	1	
Cau 5:	6	
Cau 6:	14	61
Cau 7:	2 3 5	
Cau 8:	3	0
Cau 9:	6	
Cau 10:		0
Cau 11:		

BT2.

Từ 1 đến  $n$  (với  $n \leq 1$  tỷ) có bao nhiêu số nguyên tố ?

BT3.

Tìm các số  $p, q < M$  (với  $M \leq 1$  triệu,  $p$  khác  $q$ ) sao cho tổng các ước số không kể chính nó của  $p$  bằng  $q$  và tổng các ước số thực sự của  $q$  bằng  $p$  (ví dụ 6 có các ước 1,2,3).

**BT4.**Viết hàm tìm ước số chung lớn nhất của hai số tự nhiên  $a, b$ .

**BT5.**Cho số nguyên dương  $n$ , hãy viết các hàm thực hiện các công việc sau:

a.In các chữ số của  $n$  theo chiều ngược lại;

b.Tìm chữ số lớn nhất của số  $n$ ;

c.Tính tổng các chữ số của số  $n$ ;

d.Đếm số lượng các chữ số của số  $n$ .

**BT6.**Viết hàm chuyển đổi một số  $n$  trong hệ đếm thập phân thành số trong hệ đếm cơ số  $b$ .

**BT7.**Cho số nguyên dương  $n$ .

Tính tổng  $s = 1 + 1 \times 2 + 1 \times 2 \times 3 + \dots + 1 \times 2 \times 3 \times \dots \times n$ .

**BT8.**Cho số nguyên dương  $n$ .

Tính tổng  $S = 1 \times 2 + 2 \times 3 \times 4 + \dots + n \times (n+1) \times \dots \times (2n)$ .

**BT9** Với mỗi  $n \geq 1$ , dãy số  $Y_n$  được định nghĩa như sau :

$$Y_1 = 1, Y_2 = 2, Y_3 = 3,$$

$$Y_n = Y_{n-1} + 2Y_{n-2} + 3Y_{n-3} \text{ với mọi } n \geq 4.$$

a. Hãy viết hàm tính  $Y_n$  bằng cách sử dụng đệ quy.

b. Hãy viết hàm tính  $Y_n$  bằng cách không sử dụng đệ quy.

**BT10** Cho dãy số  $A_n$  ( $n$  là số nguyên dương) được biểu diễn theo công thức đệ quy sau:

$$A_1 = 1;$$

$$A_2 = 2;$$

$$A_3 = 3;$$

$$A_n = 24A_{n-1} + 25A_{n-2} + 10A_{n-3} + 2018 \text{ mọi } n > 3.$$

a. Viết hàm tính  $A_n$  bằng cách sử dụng đệ quy.

b. Viết hàm tính  $A_n$  bằng cách không sử dụng đệ quy và cũng không sử dụng cấu trúc dữ liệu mảng.

BT1.1 Dãy  $x_n$  được định nghĩa như sau:

$$x_1 = 1;$$

$$x_n = n(x_1 + x_2 + x_3 + \dots + x_{n-1}), \forall n > 1.$$

- a. Hãy cho biết giá trị  $x_7$ .
- b. Viết hàm tính giá trị của  $x_n$ .

BT1.2 Dãy số  $x_n$  được định nghĩa như sau:

$$x_0 = 1; x_1 = 1;$$

$$x_n = nx_0 + (n-1)x_1 + (n-2)x_2 + (n-3)x_3 + \dots + x_{n-1}, \forall n > 1.$$

- a. Hãy cho biết giá trị  $x_7$ .
- b. Viết hàm tính giá trị  $x_n$ .

BT1.3

Cho  $f_1=1; f_2=1$ ; và  $f_n=f_{n-1}+f_{n-2}$  với mọi  $n > 2$ , và  $S_n$  được cho bởi công thức sau:

$$S_n = \frac{1}{1+f_1} + \frac{2}{1+f_2} + \dots + \frac{n}{1+f_n} \text{ (với } n \text{ là số nguyên dương).}$$

- a. Hãy cho biết giá trị  $S_5$
- b. Hãy viết hàm tính  $S_n$ .