

# Ngôn ngữ lập trình C

## B5: Các phép toán trong C



**PHENIKAA**  
UNIVERSITY

**Khoa Công nghệ thông tin**

# Chủ đề

- Các phép toán trong C
  - Nhóm các phép toán số học.
  - Nhóm các phép toán thao tác trên bit.
  - Nhóm các phép toán quan hệ.
  - Nhóm các phép toán logic.
  - Ngoài ra C còn cung cấp một số phép toán khác nữa như phép gán, phép lấy địa chỉ...
  - Bài tập thực hành.

# Phép toán số học

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
-	Phép đổi dấu	Số thực hoặc số nguyên	<code>int a, b;</code> <code>-12; -a; -25.6;</code>
+	Phép toán cộng	Số thực hoặc số nguyên	<code>float x, y;</code> <code>5 + 8; a + x;</code> <code>3.6 + 2.9;</code>
-	Phép toán trừ	Số thực hoặc số nguyên	<code>3 - 1.6; a - 5;</code>
*	Phép toán nhân	Số thực hoặc số nguyên	<code>a * b; b * y;</code> <code>2.6 * 1.7;</code>
/	Phép toán chia	Số thực hoặc số nguyên	<code>10.0/3.0; (bằng 3.33...)</code> <code>10/3.0; (bằng 3.33...)</code> <code>10.0/3; (bằng 3.33...)</code>
/	Phép chia lấy phần nguyên	Giữa 2 số nguyên	<code>10/3; (bằng 3)</code>
%	Phép chia lấy phần dư	Giữa 2 số nguyên	<code>10%3; (bằng 1)</code>

# Phép toán trên bits

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ	
&	Phép VÀ nhị phân	2 số nhị phân	0 & 0	(có giá trị 0)
			0 & 1	(có giá trị 0)
			1 & 0	(có giá trị 0)
			1 & 1	(có giá trị 1)
			101 & 110	(có giá trị 100)
	Phép HOẶC nhị phân	2 số nhị phân	0   0	(có giá trị 0)
			0   1	(có giá trị 0)
			1   0	(có giá trị 0)
			1   1	(có giá trị 1)
			101   110	(có giá trị 111)

# Phép toán trên bits

^	Phép HOẶC CÓ LOẠI TRỪ nhị phân	2 số nhị phân	$0 \wedge 0$	(có giá trị 0)
			$0 \wedge 1$	(có giá trị 1)
			$1 \wedge 0$	(có giá trị 1)
			$1 \wedge 1$	(có giá trị 0)
			$101 \wedge 110$	(có giá trị 011)
<<	Phép DỊCH TRÁI nhị phân	Số nhị phân	$a \ll n$	(có giá trị $a \cdot 2^n$ )
			$101 \ll 2$	(có giá trị 10100)
>>	Phép DỊCH PHẢI nhị phân	Số nhị phân	$a \gg n$	(có giá trị $a / 2^n$ )
			$101 \gg 2$	(có giá trị 1)
~	Phép ĐẢO BIT nhị phân (lấy Bù 1)	Số nhị phân	$\sim 0$	(có giá trị 1)
			$\sim 1$	(có giá trị 0)
			$\sim 110$	(có giá trị 001)

# Phép toán quan hệ

Toán tử	Ý nghĩa	Ví dụ
$>$	So sánh lớn hơn giữa 2 số nguyên hoặc thực.	$2 > 3$ (có giá trị 0) $6 > 4$ (có giá trị 1) $a > b$
$>=$	So sánh lớn hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$6 >= 4$ (có giá trị 1) $x >= a$
$<$	So sánh nhỏ hơn giữa 2 số nguyên hoặc thực.	$5 < 3$ (có giá trị 0),
$<=$	So sánh nhỏ hơn hoặc bằng giữa 2 số nguyên hoặc thực.	$5 <= 5$ (có giá trị 1) $2 <= 9$ (có giá trị 1)
$==$	So sánh bằng nhau giữa 2 số nguyên hoặc thực.	$3 == 4$ (có giá trị 0) $a == b$
$!=$	So sánh không bằng (so sánh khác) giữa 2 số nguyên hoặc thực.	$5 != 6$ (có giá trị 1) $6 != 6$ (có giá trị 0)

# Phép toán logic

Toán tử	Ý nghĩa	Kiểu dữ liệu của toán hạng	Ví dụ
&&	Phép VÀ LOGIC. Biểu thức VÀ LOGIC bằng 1 khi và chỉ khi cả 2 toán hạng đều bằng 1	Hai biểu thức logic	$3 < 5 \ \&\& \ 4 < 6$ (có giá trị 1) $2 < 1 \ \&\& \ 2 < 3$ (có giá trị 0) $a > b \ \&\& \ c < d$
	Phép HOẶC LOGIC. Biểu thức HOẶC LOGIC bằng 0 khi và chỉ khi cả 2 toán hạng bằng 0.	Hai biểu thức logic	$6 \    \ 0$ (có giá trị 1) $3 < 2 \    \ 3 < 3$ (có giá trị 0) $x \geq a \    \ x == 0$
!	Phép PHỦ ĐỊNH LOGIC một ngôi. Biểu thức PHỦ ĐỊNH LOGIC có giá trị bằng 1 nếu toán hạng bằng 0 và có giá trị bằng 0 nếu toán hạng bằng 1	Biểu thức logic	$!3$ (có giá trị 0) $!(2 > 5)$ (có giá trị 1)

# Phép gán

- Cú pháp:

```
tên_biến = biểu_thức;
```

- Ý nghĩa: Lấy giá trị của biểu\_thức gán cho tên\_biến.

- Ví dụ:

```
- int a, b, c;
```

```
- a = 3;
```

```
- b = a + 5;
```

```
- c = a * b;
```



# Phép gán

- Biểu thức gán là biểu thức nên nó cũng có giá trị.
- Giá trị của biểu thức gán bằng giá trị của biểu\_thức → Có thể gán giá trị của biểu thức gán cho một biến khác hoặc sử dụng như một biểu thức bình thường.
- Ví dụ:
  - `int a,b,c;`
  - `a=b=2007;`
  - `c=(a=20)*(b=30);`

# Phép gán

- Phép toán gán thu gọn:

$x = x + y;$  giống như  $x += y;$

- Dạng lệnh gán thu gọn này còn áp dụng được với các phép toán khác:

$+, -, *, /, \%, >>, <<, \&, |, ^$

# Thứ tự ưu tiên các phép toán

Mức	Các toán tử	Trật tự kết hợp
1	() [] . -> ++ (hậu tố) -- hậu tố	----->
2	! ~ ++ (tiền tố) -- (tiền tố) - *	<-----
	& sizeof	
3	* / %	----->
4	+ -	----->
5	<< >>	----->
6	< <= > >=	----->
7	== !=	----->
8	&	----->
9	^	----->
10		----->
11	&&	----->
12		----->
13	?:	<-----
14	= += -=	<-----

# Nguyên tắc xác định trật tự thực hiện các phép toán

- Biểu thức con trong ngoặc được tính toán trước các phép toán khác
- Phép toán một ngôi đứng bên trái toán hạng được kết hợp với toán hạng đi liền nó.
- Nếu toán hạng đứng cạnh hai toán tử thì có 2 khả năng là:
  - Nếu hai toán tử có độ ưu tiên khác nhau thì toán tử nào có độ ưu tiên cao hơn sẽ kết hợp với toán hạng
  - Nếu hai toán tử cùng độ ưu tiên thì dựa vào trật tự kết hợp của các toán tử để xác định toán tử được kết hợp với toán hạng.

# Các loại biểu thức

- Biểu thức số học:
  - Là biểu thức mà giá trị của nó là các đại lượng số học (số nguyên, số thực).
  - Các toán tử là các phép toán số học (cộng, trừ, nhân, chia...).
  - Các toán hạng là các đại lượng số học (số, biến, hằng).
- Ví dụ:
  - $a, b, c$  là các biến thuộc một kiểu dữ liệu số nào đó.
  - $3 * 3.7$
  - $8 + 6 / 3$
  - $a + b - c \dots$

# Các loại biểu thức

- Biểu thức logic:
  - Là biểu thức mà giá trị của nó là các giá trị logic, tức là một trong hai giá trị: Đúng (TRUE) hoặc Sai (FALSE).
    - Giá trị nguyên khác 0: Đúng (TRUE),
    - Giá trị 0: Sai (FALSE).
- Các phép toán logic gồm có:
  - AND: VÀ logic, kí hiệu là &&
  - OR: HOẶC logic, kí hiệu là ||
  - NOT: PHỦ ĐỊNH, kí hiệu là !

# Các loại biểu thức

- Ví dụ về biểu thức logic:

- `(5 > 7) && (9!=10)`      • `// có giá trị logic là sai, FALSE`
- `0 || 1`      • `// có giá trị logic là đúng, TRUE`
- `(5 > 7) || (9!=10)`      • `// có giá trị logic là đúng, TRUE`
- `0`      • `// có giá trị logic là sai, FALSE`
- `!0`      • `// phủ định của 0, có giá trị logic là đúng, TRUE`
- `3`      • `// có giá trị logic là đúng, TRUE`
- `!3`      • `// phủ định của 3, có giá trị logic là sai, FALSE`
- `(a > b) && ( a < b)`      • `// Có giá trị sai, FALSE. Giả sử a, b là 2 biến kiểu int`

# Các loại biểu thức

- Biểu thức quan hệ:
    - Là những biểu thức trong đó có sử dụng các toán tử quan hệ so sánh như lớn hơn, nhỏ hơn, bằng nhau, khác nhau, ...
    - Chỉ có thể nhận giá trị là một trong 2 giá trị Đúng (TRUE) hoặc Sai (FALSE).
- Biểu thức quan hệ là một trường hợp riêng của biểu thức logic.



# Các loại biểu thức

- Ví dụ về biểu thức quan hệ:

- $5 > 7$

- $9 \neq 10$

- $2 \geq 2$

- $a > b$

- $a+1 > a$

- *// có giá trị logic là sai, FALSE*

- *// có giá trị logic là đúng, TRUE*

- *// có giá trị logic là đúng, TRUE*

- *// giả sử a, b là 2 biến kiểu int*

- *// có giá trị đúng, TRUE*

# Sử dụng biểu thức

- Làm vế phải của lệnh gán.
- Làm toán hạng trong các biểu thức khác.
- Làm tham số thực trong lời gọi hàm.
- Làm chỉ số trong các cấu trúc lặp `for`, `while`, `do while`.
- Làm biểu thức kiểm tra trong các cấu trúc rẽ nhánh `if`, `switch`.

# Một số toán tử đặc trưng

- Phép toán lấy địa chỉ biến (&):

`&<tên biến>;`

- Các phép toán tăng giảm một đơn vị: Tăng hoặc giảm một đơn vị cho biến.

`<tên biến> = <tên biến> + 1;`

$\Leftrightarrow$  `<tên biến>++`

`<tên biến> = <tên biến> - 1;`

$\Leftrightarrow$  `<tên biến>--`

- Ví dụ:

`- int a = 5;`

`- float x = 10;`

`- a++; // tương đương với a = a + 1;`

`- x--; // tương đương với x = x - 1;`

# Một số toán tử đặc trưng

- Các phép toán tăng giảm một đơn vị:
  - Tiền tố: Thay đổi giá trị của biến trước khi sử dụng.
  - Hậu tố: Tính toán giá trị của biểu thức bằng giá trị ban đầu của biến, sau đó mới thay đổi giá trị của biến

- Ví dụ:

```
int a, b, c;
```

```
a=3; // a bằng 3
```

```
b=a++; // Dạng hậu tố: b bằng 3, a bằng 4
```

```
c=++b; // Dạng tiền tố: b bằng 4, c bằng 4
```

# Một số toán tử đặc trưng

- Phép toán chuyển đổi kiểu bắt buộc:  
`(<kiểu dữ liệu mới>) <biểu thức>;`
- Chương trình dịch sẽ tự động chuyển đổi kiểu:
  - Số nguyên: `int`  $\rightarrow$  `long int`
  - Số: `long int`  $\rightarrow$  `float`
- Ngược lại
  - Số nguyên `long int` 50,000 không phải là một số nguyên kiểu `int` vì phạm vi biểu diễn của kiểu `int` là từ (-32,768 đến 32,767)  $\rightarrow$  **Phải ép kiểu**
- C hỗ trợ chuyển kiểu tự động trong những trường hợp sau

`char`  $\rightarrow$  `int`  $\rightarrow$  `long int`  $\rightarrow$  `float`  $\rightarrow$  `double`  $\rightarrow$  `long double`

# Biểu thức điều kiện

- Cú pháp:

`biểu_thức_1 ? biểu_thức_2 : biểu_thức_3`

- Giá trị của biểu thức điều kiện:

- Nhận giá trị của `biểu_thức_2` nếu `biểu_thức_1` có giá trị khác 0 (tương ứng với giá trị logic ĐÚNG),
- Ngược lại: Giá trị của `biểu_thức_3` nếu `biểu_thức_1` có giá trị bằng 0 (tương ứng với giá trị logic SAI).

- Ví dụ:

```
float x, y, z;    // khai báo biến
x = 3.8; y = 2.6; // gán giá trị cho các biến x, y
z = (x < y) ? x : y; // z sẽ có giá trị bằng giá trị
                    // nhỏ nhất trong 2 số x và y
```

# BÀI TẬP

## 1. Chương trình sau in ra kết quả gì?

```
1  #include <stdio.h>
2  int main()
3  {
4      int a = -2, b = 2, c, d;
5      c = ((a+b) == !b);
6      a = -((a < c) || (--b == c));
7      d = (!(a != -c) && (b+a >= c) && ((a+c)%2 || (-a+c-b)));
8      printf("Ket qua d=%d\n",d);
9      return 0;
10 }
```

# BÀI TẬP

2. Tính giá trị của biểu thức sau và đưa ra kết quả ra màn hình cùng với giá trị của biến a, b, c (với a, b, c là các số nguyên nhập vào từ bàn phím) sau khi thực hiện phép toán:

$$Kq1 = a++ + ++a$$

$$Kq2 = --a - b-- * ++c$$

3. Dùng biểu thức điều kiện ?: để đưa ra số lớn nhất trong 3 số thực nhập vào từ bàn phím.



# BÀI TẬP

4. Viết chương trình đọc lượng nước tiêu thụ theo mét khối và tính số tiền phải trả theo bảng sau.

Ví dụ: nếu mức tiêu thụ là 35cm thì chi phí cuối cùng được tính như sau: chi phí =  $(30-5)*1,2 + (35-30)*1,5$ .

<i>CONSUMPTION (CM)</i>	<i>COST (CM)</i>
[0–5]	0
(5–30]	1.2
> 30	1.5

# BÀI TẬP

5. Nhập từ bàn phím vào 3 số thực  $a, b, c$ . Tính delta theo công thức sau:  $b^2 - 4.a.c$  rồi đưa kết quả ra màn hình.

6. Viết chương trình giải phương trình bậc 2:

$$a.x^2 + b.x + c = 0$$

Với  $a, b, c$  là các số thực nhập từ bàn phím sao cho  $\text{delta} > 0$ .

7. Tính giá trị của biểu thức sau:

$$F = \frac{x + y + \sqrt{z}}{x^2 + y^2 + 1} - |\sin(x) - z \cos(y)|$$

Với  $x, y, z$  là các số thực bất kỳ nhập từ bàn phím. Gợi ý: sử dụng các hàm trong thư viện `math.h` như: `sqrt`: tính căn bậc 2; `sin/cos`: để tính sin/cos; sử dụng thư viện `stdlib.h` có hàm `abs/fabs`: tính giá trị tuyệt đối của số nguyên/thực;