TOÁN TỬ VÀ BIỂU THỨC

# Lý thuyết:

## Toán tử số học trong C

Bảng dưới đây chỉ ra tất cả các toán tử số học được hỗ trợ bởi ngôn ngữ C. Giả sử biến **A** có giá trị 10 và biến **B** có giá trị 20:

[**Hiển thị ví dụ**](http://vietjack.com/lap_trinh_c/toan_tu_so_hoc_trong_c.jsp)

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Miêu tả** | **Ví dụ** |
| + | Thêm hai toán hạng | A + B sẽ cho kết quả là 30 |
| - | Trừ giá trị toán hạng hai từ toán hạng đầu | A - B sẽ cho kết quả là -10 |
| \* | Nhân hai toán hạng | A \* B sẽ cho kết quả là 200 |
| / | Chia lấy phần nguyên hai toán hạng | B / A sẽ cho kết quả là 2 |
| % | Chia lấy phần dư | B % A sẽ cho kết quả là 0 |
| ++ | Lượng gia giá trị toán hạng thêm 1 đơn vị | A++ sẽ cho kết quả là 11 |
| -- | Lượng giảm giá trị toán hạng một đơn vị | A-- sẽ cho kết quả là 9 |

main()

{

int x,y;

x = 5;

y = 2;

printf("The integers are : %d & %d\n", x, y);

printf("The addition gives : %d\n", x + y);

printf("The subtraction gives : %d\n", x - y);

printf("The multiplication gives : %d\n", x \* y);

printf("The division gives : %d\n", x / y);

printf("The modulus gives : %d\n", x % y);

getchar();

}

Kết quả là:

The integers are : 5 & 2

The addition gives : 7

The subtraction gives : 3

The multiplication gives : 10

The division gives : 2

The modulus gives : 1

## Toán tử quan hệ

Toán tử quan hệ được dùng để kiểm tra mối quan hệ giữa hai biến, hay giữa một biến và một hằng.

Bảng sau mô tả ý nghĩa của các toán tử quan hệ.

|  |  |
| --- | --- |
| **Toán tử** | **Ý nghĩa** |
| > | lớn hơn |
| >= | lớn hơn hoặc bằng |
| < | nhỏ hơn |
| <= | nhỏ hơn hoặc bằng |
| **==** | bằng |
| != | không bằng |

**Bảng Toán tử quan hệ và ý nghĩa**

**Toán tử luận lý (Logical Operators)**

Toán tử luận lýlà các ký hiệu dùng để kết hợp hay phủ định biểu thức có chứa các toán tử quan hệ.

Những biểu thức dùng toán tử luận lý trả về **0** cho **false** và **1** cho **true**

|  |  |
| --- | --- |
| **Toán tử** | **Ý nghĩa** |
| && | AND: trả về kết quả là **true** khi cả 2 toán hạng đều **true** |
| || | OR : trả về kết quả là **true** khi chỉ một trong hai toán hạng đều **true** |
| ! | NOT: Chuyển đổi giá trị của toán hạng duy nhất từ **true** thành **false** và ngược lại. |

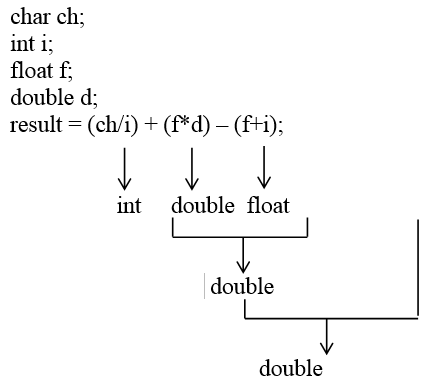
**Toán tử luận lý và ý nghĩa**

Lưu ý: *Bất cứ toán tử luận lý nào có ký hiệu là hai ký tự thì không được có khoảng trắng giữa hai ký tự đó*, ví dụ : **==** sẽ không đúng nếu viết là = =.

**Biểu thức dạng hỗn hợp & Chuyển đổi kiểu**

Một biểu thức dạng hỗn hợp là một biểu thức mà trong đó các toán hạng của một toán tử thuộc về nhiều kiểu dữ liệu khác nhau. Những toán hạng này thông thường được chuyển về cùng kiểu với toán hạng có kiểu dữ liệu lớn nhất. Điều này được gọi là tăng cấp kiểu. Sự phát triển về kiểu dữ liệu theo thứ tự sau :

char < int <long <float <double



**Ép kiểu (Casts)**

Thông thường, ta nên đổi tất cả hằng số nguyên sang kiểu float nếu biểu thức bao gồm những phép tính số học dựa trên số thực, nếu không thì vài biểu thức có thể mất đi giá trị thật của nó.Ta xem ví dụ:

int x,Y;

float z;

x = 10;

y = 100;

z = x/y;

Trong trường hợp này, **z** sẽ được gán **0** khi phép chia diễn ra và phần thập phân (0.10) sẽ bị cắt bỏ.

Do đó một biểu thức có thể được ép thành một kiểu nhất định. Cú pháp chung của **cast** là:

(kiểu dữ liệu) biểu thức

Ví dụ, để đảm bảo rằng biểu thức **a/b,** với **a** và **b** là số nguyên, cho kết quả là kiểu **float,** dòng mã sau được viết:

(float) a/b;

**Độ ưu tiên của toán tử**

Thứ tự ưu tiên của toán tử số học được thể hiện như bảng dưới đây.

|  |  |  |
| --- | --- | --- |
| **Loại toán tử** | **Toán tử** | **Tính kết hợp** |
| Một ngôi | - , ++, -- | Phải sang trái |
| Hai ngôi | ^ | Trái sang phải |
| \*, /, % |
| +, - |
| = | Phải sang trái |

thứ tự ưu tiên cho toán tử luận lý.

| **Thứ tự** | **Toán tử** |
| --- | --- |
| 1 | NOT |
| 2 | AND |
| 3 | OR |

Bảng 4.5: Thứ tự ưu tiên cho toán tử luận lý

*Thứ tự ưu tiên giữa các kiểu toán tử khác nhau:*

| **Thứ tự** | **Kiểu toán tử** |
| --- | --- |
| 1 | Số học |
| 2 | So sánh (Quan hệ) |
| 3 | Luận lý |

**Xét ví dụ sau:**

2\*3+4/2 > 3 AND 3<5 OR 10<9

Việc thực hiện tính toán sẽ như sau:

1. [2\*3+4/2] > 3 AND 3<5 OR 10<9  
   Ðầu tiên toán tử số học sẽ được tính theo thứ tự ưu tiên như bảng 4.4.
2. [[2\*3]+[4/2]] > 3 AND 3<5 OR 10<9
3. [6+2] >3 AND 3<5 OR 10<9
4. [8 >3] AND [3<5] OR [10<9]  
   Kế đến sẽ tính tất cả những toán tử so sánh có cùng độ ưu tiên theo quy tắc tính từ trái sang phải.
5. True AND True OR False  
   Cuối cùng tính toán các toán tử kiểu luận lý. AND sẽ có độ ưu tiên cao hơn OR.
6. [True AND True]OR False  
   True OR False
7. True

**Thực hành với biểu thức sau:**

5+9\*3^2-4 > 10 AND (2+2^4-8/4 > 6 OR (2<6 AND 10>11))