

運算與表達式

Hanly 《C 語言詳論》6e

§ 1 名詞速查表

中文	English	一句話定義	科目	章節
運算子	operator	對運算元做運算、產生結果的符號，如 <code>+</code> 、 <code>==</code> 、 <code>&&</code>	程式語言一	運算與表達式
運算元	operand	被運算子作用的值，如 <code>7 + 3</code> 的 7 和 3	程式語言一	運算與表達式
取餘數	modulo	<code>%</code> ，回傳整數除法的餘數，如 <code>7 % 3 = 1</code>	程式語言一	運算與表達式
關係運算子	relational operator	比較大小，結果是 1（真）或 0（假）	程式語言一	運算與表達式
邏輯運算子	logical operator	<code>&&</code> 、 <code> </code> 、 <code>!</code> ，組合條件	程式語言一	運算與表達式
短路求值	short-circuit	左邊已能決定結果時，右邊不執行	程式語言一	運算與表達式
遞增/遞減	increment/ decrement	<code>++</code> / <code>--</code> ，變數加 1/減 1；分前綴與後綴	程式語言一	運算與表達式
運算子優先序	precedence	決定先算哪個運算子（如 <code>*</code> 先於 <code>+</code> ）	程式語言一	運算與表達式
型別提升	type promotion	混合型別運算時自動轉成較大的型別	程式語言一	運算與表達式

§ 2 核心概念

核心概念

運算子對運算元做運算、產生結果。C 的關係 (`==`、`>`…) 與邏輯 (`&&`、`||`、`!`) 運算結果是**整數 1 (真) 或 0 (假)**，可以當數字繼續算。C 沒有原生 `bool` (C99 才有 `_Bool`)：**任何非零值都算真、0 算假**。

兩個最致命的混淆：`==` (比較) 不是 `=` (指派)；整數相除是整數除法 (`7/3 = 2`，不是 `2.33`)。

§ 3 主要內容

3.1 算術運算子與整數除法

`+` `-` `*` `/` 與取餘 `%`。**整數除法會截斷**：`7 / 3 = 2`；要小數得用浮點 `7.0 / 3` 或 `(double)7 / 3`。
`%` 只能用在整數。

`%` 的常見用途：判奇偶 (`n % 2`)、取個位 (`n % 10`)、循環計數 (`i % 3` 得 `0,1,2,0...`)、秒轉分秒 (`500/60=8` 分、`500%60=20` 秒)。拆數字：`(num / 10) % 10` 取十位。

負數的除法 (C99)：**除法向零截斷、餘數符號跟被除數一致**。驗證公式 `商 × 除數 + 餘 = 被除數` 永遠成立。例：`-7 / 3 = -2`、`-7 % 3 = -1` (`-2×3 + (-1) = -7`)。

3.2 關係運算子：結果是 0 或 1

`==` `!=` `>` `<` `>=` `<=`，真給 1、假給 0，**可以當整數算**：`(a>b) + (a==5)` 是合法算術。

`==` **vs** `=` 是最經典的 **bug**：`if (x = 3)` 是把 3 指派給 `x` (回傳 3，非零=真，**永遠執行**)；要比較得寫 `if (x == 3)`。防禦寫法 `if (3 == x)` (誤寫成 `=` 會編譯錯)。

`3 > 2 > 1` **不是數學連續不等式**：由左到右算 `(3>2)=1`、再 `(1>1)=0`，結果是 0。要表達連續比較得寫 `3 > 2 && 2 > 1`。

3.3 邏輯運算子與短路求值

`&&` (AND，全真才真)、`||` (OR，有真就真)、`!` (NOT，反轉)。

短路求值：左邊已能決定結果時，右邊**不執行**。`x != 0 && 10/x > 2` —— `x==0` 時右邊不算，**避免除以零**；這不只是效能，更是安全機制 (也常用來避開空指標)。

3.4 遞增遞減：前綴 vs 後綴

`++` / `--` 讓變數加 1 / 減 1。 `i++` (後綴) 先用舊值再加； `++i` (前綴) 先加再用。單獨一行時兩者相同；放在運算式裡差很多：

```
int i = 5, a = i++; // a = 5 (先用), 之後 i = 6
int j = 5, b = ++j; // b = 6 (先加), 之後 j = 6
```

3.5 優先序：不確定就加括號

大致順序：`*` / `%` 高於 `+` - 高於關係 (`<` `>` `==` ...) 高於 `&&` 高於 `||` 高於指派 `=`。 `a + b * c` 先乘； `a < b && c < d` 先各自比較再 AND。 **不確定就加括號**，可讀又安全。

3.6 混合型別：型別提升

運算式混合型別時自動轉成較大的：`int + double` → `double`。所以 `(double)7 / 3` 或 `7.0 / 3` 會得浮點結果，而 `7 / 3` 兩個 `int` 是整數除法。

§ 4 語法與函式速查

算術	<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>%</code> (只整數)	關係	<code>==</code> <code>!=</code> <code>></code> <code><</code> <code>>=</code> <code><=</code> <code>→</code> <code>1/0</code>
邏輯	<code>&&</code> <code> </code> <code>!</code> (非零=真, 0=假)	指派	<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code> <code>%=</code>
遞增遞減	<code>i++</code> <code>++i</code> <code>i--</code> <code>--i</code>	優先序	<code>*</code> <code>/</code> <code>%</code> <code>></code> <code>+</code> <code>-</code> <code>></code> 關係 <code>></code> <code>&&</code> <code>></code> <code> </code>

§ 5 常見錯誤

常見錯誤

- `if (x = 3)` 把 `==` 寫成 `=` (變指派、永遠真)。
- 整數除法忘了截斷：`7/3=2`，要小數得轉浮點。
- `3 > 2 > 1` 當成連續不等式 (其實由左到右算、結果 0)。
- 混淆 `i++` 與 `++i` 在運算式中的值。
- 浮點用 `==` 比較 (用容差)。
- 對負數的 `%` 結果符號判斷錯 (餘數跟被除數同號)。
- 優先序記不熟又不加括號。

§ 6 練習題

練習 1 (一般題) : % 拆數字

寫出輸出。

```
int num = 1234;  
printf("%d %d\n", num % 10, (num / 10) % 10);  
printf("%d\n", 7 % 3);
```

引導步驟

1. $\% 10$ 取最低位； $(\text{num}/10)\%10$ 先去個位再取最低位 = 十位。

解答

```
4 3  
1
```

$1234 \% 10 = 4$ (個位) 、 $(1234/10)\%10 = 123\%10 = 3$ (十位) ; $7 \% 3 = 1$ 。

練習 2 (一般題)：關係與短路

寫出各輸出。

```
int a = 5, b = 3;
printf("%d\n", (a > b) + (a == 5) + (b != 3));
printf("%d\n", 3 > 2 > 1);
```

引導步驟

1. 關係結果是 0/1，可相加。
2. $3 > 2 > 1$ 由左到右： $(3 > 2) = 1$ ，再 $1 > 1$ 。

解答

```
2
0
```

$(5 > 3) + (5 == 5) + (3 != 3) = 1 + 1 + 0 = 2$ ； $3 > 2 > 1 = (1) > 1 = 0$ (非連續比較)。

§ 7 自我檢核

- 知道整數除法會截斷，`%` 只能用在整數。
- 會用 `%` 判奇偶、取個位、拆數字。
- 知道關係/邏輯結果是 0/1，可當整數算。
- 分得清 `==`（比較）與 `=`（指派），知道 `if(x=3)` 的陷阱。
- 知道 `3>2>1` 不是連續比較，要用 `&&`。
- 懂短路求值，能用它避免除以零。
- 分得清 `i++` 與 `++i` 在運算式中的值。
- 記得運算優先序，不確定就加括號。