

3-1 運算式與運算子

重點整理

1. **運算式(expression)**是由**運算子(operator)**和**運算元(operand)**所組成：

- (1) 運算子：負責運算的加、減、乘、除等符號。
- (2) 運算元：運算執行的對象，包括字面值、變數或其它運算式等。

2. 運算子的種類：

(1) 算術運算子

運算子	說明	範例
+	加法	8+5，結果為 13
-	減法	8-5，結果為 3
*	乘法	8*5，結果為 40
/	除法	8/5，結果為 1(當運算元皆為整數時，會捨去小數部份) 8.0/5.0，結果為 1.6(當運算元為浮點數時)
%	餘數	8%5，結果為 3 商: 1 餘數:3
+	正號	+5，結果為正 5
-	負號	-5，結果為負 5

(2) 指定運算子，將右邊運算式的結果，指定給左邊

運算子	說明	範例
=	指定	x=10 x=a+b
+=	加法後指定	x+=5，結果與 x=x+5 相同
-=	減法後指定	x-=5，結果與 x=x-5 相同
=	乘法後指定	x=5，結果與 x=x*5 相同
/=	除法後指定	x/=5，結果與 x=x/5 相同
%=	餘數後指定	x%=5，結果與 x=x%5 相同

a = b+c+5;
運算子: =,+
運算元: a,c,b

- (3) 遞增或遞減運算子：分為前置與後置兩種用法，在程式敘述句中，若遞增(遞減)運算子為前置，則一定最先被執行，若遞增(遞減)運算子為後置，則一定最後被執行。

運算子	說明	範例
++	遞增	<p>前置：++x，對變數 x 的值加 1，結果與 $x=x+1$ 相同。</p> <p>後置：x++，對變數 x 的值加 1，結果與 $x=x+1$ 相同。</p> <p>若變數 a 和 b 的值皆為 5，</p> <p>a=++b; 先將 b 的值加 1，再將 b 指定給 a，執行後 a 為 6，b 為 6。</p> <p>a=b++; 先將 b 的值指定給 a，b 再加 1，執行後 a 為 5，b 為 6。</p> <p>printf("%d", ++a); 先將 a 的值加 1 後再輸出，執行後 a 的值為 6，螢幕輸出 6。</p> <p>printf("%d", b++); 先將 b 輸出後，再將 b 的值加 1，執行後 b 的值為 6，螢幕輸出 5。</p> <p>if(a==++b){ } 先將 b 的值加 1，再進行條件判斷，執行後 b 的值為 6，條件式($a==b$)不成立。</p> <p>if(a==b++){ } 先進行條件判斷，再將 b 的值加 1，執行後 b 的值為 6，條件式($a==b$)成立。</p>
--	遞減	<p>前置：--x，對變數 x 的值減 1，結果與 $x=x-1$ 相同。</p> <p>後置：x--，對變數 x 的值減 1，結果與 $x=x-1$ 相同。</p> <p>若變數 a 和 b 的值皆為 5，</p> <p>a--b; 先將 b 的值減 1，再將 b 指定給 a，執行後 a 為 4，b 為 4。</p> <p>a=b--; 先將 b 的值指定給 a，b 再減 1，執行後 a 為 5，b 為 4。</p> <p>printf("%d", --a); 先將 a 的值減 1 後再輸出，執行後 a 的值為 4，螢幕輸出 4。</p> <p>printf("%d", b--); 先將 b 輸出後，再將 b 的值減 1，執行後 b 的值為 4，螢幕輸出 5。</p> <p>if(a==--b){ } 先將 b 的值減 1，再進行條件判斷，執行後 b 的值為 4，條件式($a==b$)不成立。</p> <p>if(a==b--){ } 先進行條件判斷，再將 b 的值減 1，執行後 b 的值為 4，條件式($a==b$)成立。</p>

(4) 關係運算子：用來建立條件運算式。

運算子	說明	範例
==	等於	5 == 5，條件成立，結果為真(true)。 5 == 6，條件不成立，結果為假(false)。
!=	不等於	5 != 6，條件成立，結果為真。 5 != 5，條件不成立，結果為假。
>	大於	6 > 5，條件成立，結果為真。 5 > 6，條件不成立，結果為假。
<	小於	5 < 6，條件成立，結果為真。 6 < 5，條件不成立，結果為假。
>=	大於等於	6 >= 5，條件成立，結果為真。 6 >= 6，條件成立，結果為真。 5 >= 6，條件不成立，結果為假。
<=	小於等於	5 <= 6，條件成立，結果為真。 6 <= 6，條件成立，結果為真。 6 <= 5，條件不成立，結果為假。

(5) 邏輯運算子：可以結合多個條件運算式。

運算子	說明	範例
&&	且(AND)，所有條件都成立，結果才為真(true)。	(5 == 5) && (6 == 6)，結果為真(true)。 (5 == 6) && (6 == 6)，結果為假(false)。
	或(OR)，只要其中一個條件成立，結果就為真(true)。	(5 == 6) (6 == 6)，結果為真。 (5 == 6) (6 == 7)，結果為假。
!	非(NOT)，條件不成立，視為真(true)。	!(5 == 6)，結果為真。 !(5 == 5)，結果為假。

註 C 語言沒有布林資料型態，而是使用「0」表示 false，「非 0」表示 true，而條件運算式的結果若為 false 回傳值為「0」，若為 true，則回傳的值為「1」。

(6) 位元運算子：針對運算元的每一個位元進行運算。

運算子	說明	範例
~	NOT(1 的補數)	~1010，結果為 0101。
&	AND	1010 & 0110，結果為 0010。 <div style="text-align: right;"> 1010 0110 --- 0010 </div>
	OR	1010 0110，結果為 1110。 <div style="text-align: right;"> 1010 0110 --- 0010 </div>
^	XOR	1010 ^ 0110，結果為 1100。
<<	將所有位元左移 n 個位元，移出的位元捨棄，最低位元補 0，結果與數學運算乘 2^n 相同	00001010 << 2，左移 2 個位元，結果為 00101000(乘 2^2)。 <div style="text-align: center;"> </div>
>>	將所有位元右移 n 個位元，移出的位元捨棄，原最高位元為 0 則補 0，為 1 則補 1，結果與數學運算除 2^n 相同	00001100 >> 2，右移 2 個位元，結果為 00000011(除 2^2)。 <div style="text-align: center;"> </div> 11110100 >> 2，右移 2 個位元，結果為 11111101(除 2^2)。 <div style="text-align: center;"> </div>

(7) 其它運算子。

運算子	說明	範例
sizeof	(1) 取得資料型態所需的記憶體空間。 (2) 取得變數佔用的記憶體空間。	(1) char 型態的大小為 1Byte，double 型態的大小為 8Byte，所以 sizeof(char)的結果為 1，sizeof(double)的結果為 8。 (2) 若宣告變數 a 為 char 型態，變數 b 為 double 型態，則 sizeof(a)的結果為 1，sizeof(b)的結果為 8。
(cast type)	型態轉換運算子 3-2節說明	將變數的值或運算式的結果，強制轉換其資料型態

C 語言強制轉型 (casting)

char to int 與 short to int
 int to float
 int to unsigned int
 int and const int

<https://hackmd.io/@chenishi/rJJfgHEa7?type=view>

3. 如同數學運算中的先乘除後加減，運算子也有其優先順序，如下：

優先順序	運算子
1	()
2	正號(+)、負號(-)、++、--、!、~、sizeof、(cast type)
3	*、/、%
4	加法(+)、減法(-)
5	<<、>>
6	>、<、>=、<=
7	==、!=
8	&
9	^
10	
11	&&
12	
13	=、+=、-=、\=、%=

4. 若運算子只需要 1 個運算元時，稱為一元運算子，如++、--、!、^等。 ++a

5. 若運算子需要 2 個運算元時，稱為二元運算子，如%、&&、>>、==等。 c=a+b

01 範例

在 C 語言中宣告 6 個整數變數 a、b、c、d、e、f，其值分別為 1、2、3、4、5、6，請問下列的運算式，其運算後 f 的值為何？

(1) $f = c + b * (e - c) / b$

(2) $f = -f - a + b * (-d);$

(3) $f = c / b + e \% c;$

(4) $f = b * (d++);$

1. $f = b * (d) = 2 * 4 = 8$
2. $d = d + 1 = 4 + 1 = 5$

(5) $f = c + (++e) - (--d);$

(6) $f \% = e;$

(7) $f += (a++) * (c--) + (++e);$

1. $e = e + 1$
2. $d = d - 1$
3. $f = c + e - d$

解

(1) 與一般的數學運算相同，先乘除後加減，括號內先做，

$$f = 3 + 2 * (5 - 3) / 2 = 5$$

(2) $f = (-6) - 1 + 2 * (-4) = (-6) - 1 + (-8) = -15$

(3) $f = 3 / 2 + 5 \% 3 = 1 + 2 = 3$

(4) 遞增運算子雖然在小括號內，但因為是後置運算，因此會在運算式的最後才被執行，此運算式可拆解成以下兩個運算式，

$$f = b * d;$$

$$d = d + 1;$$

結果 $d = 5$ ， $f = 8$

(5) 將運算式拆解成

`e = e+1;`

`d = d-1;`

`f = c+e-d;`

結果 `d = 3` , `e = 6` , `f = 6`

(6) 等同於運算式 `f = f % e;`

`f = 6 % 5 = 1`

(7) 等同於運算式 `f = f + (a++) * (c--) + (++e);`

將運算式拆解成

`e = e+1;`

`f = f+a*c+e;`

`a = a+1;`

`c = c-1;`

結果 `a=2` , `c=2` , `e=6` , `f=15`

02 範例

下列 C 語言的程式片段執行後，變數 `x` 的值為何？

(1) `char x = 3;`

`x = x | 0x10;`

(2) `char x=3;`

`x=~x;`

(3) `char x=3;`

`x=x^0x0F;`

(4) `char x=3;`

`x=x<<2;`

(5) `char x=3;`

`x=x>>1;`

(6) `char x = -3;`

`x=x>>1;`

1. `x = 0000 0011`
 2. 反`x = 1111 1100`
 3. 負數
 4. 2's = `0000 0100`
 5. `x=-4`

1. `3 -> 0000 0011`
 2. 2's -> `1111 1101`
 3. `x>>1 -> 1111 1110`
 4. 2's -> `0000 0010`
 5. `x=-2`

解

(1) `char` 為 8 bit 的有號數，其最高位元(MSB)為符號位元(0 表示正數，1 表示負數)，若為負數則需取 2's 補數(取反相再加 1)，以得到負數的大小。

3 的 2 進制為 `00000011`，`0x10` 的 2 進制為 `00010000`，

(a) 進行 OR 運算後，得到 `00010011`

(b) 最高位元為 0，表示其為正數，運算後 `x` 的值為 19

(2) (a) `x` 取反相後為 `11111100`

(b) 最高位元為 1，表示其為負數，故取 2's 補數，得到 `00000100`，負數的大小為 4

(c) 運算後 `x` 的值為 -4

(3) `0x0F` 的 2 進制為 `00001111`

(a) 與 `x` 進行 XOR 運算後，得到 `00001100`

(b) 最高位元為 0，表示其為正數，運算後 `x` 的值為 12。

(4) `00000011` 的所有位元都左移 2 個位元，得到 `00001100`，運算後 `x` 的值為 12。

(5) `00000011` 的所有位元都右移 1 個位元，得到 `00000001`，運算後 `x` 的值為 1。

- (6) -3 以 2's 補數表示為 11111101(00000011 反相後再加 1)
- (a) 11111101 的所有位元都右移 1 個位元，因原最高位元為 1，故補 1，得到 11111110
- (b) 11111110 的最高位元為 1，表示其為負數，故取 2's 補數，得到 00000010，負數的大小為 2
- (c) 運算後 x 的值為 -2

• 立即練習

- (C) 1. 下列哪一個運算子中的優先權最高？ (A) / (B) > (C) ! (D) &&。
- (B) 2. 將運算子的優先權由高到低排序，下列何者正確？
 (A) ||、>=、%、^、=、++ (B) ++、%、>=、^、||、=
 (C) =、||、^、>=、%、++ (D) =、++、^、||、%、>=。
- (A) 3. 運算式 9/4 的結果為何？ (A) 2 (B) 0 (C) 2.25 (D) 1。
- (C) 4. 條件式 5 != 4 的回傳值為何？ (A) -1 (B) 0 (C) 1 (D) 2。
- (A) 5. 一 C 程式指令 a = 4 * 2 < 2 * 3，a 的執行結果為何？ (A) 0 (B) 1 (C) 10 (D) 12。
- (C) 6. 變數 x 為 3，則程式碼 y = ++x * 5; 執行後，y 的值為何？ (A) 15 (B) 16 (C) 20 (D) 8。
- (A) 7. 變數 x 為 3，則程式碼 y = (x++) * 5 執行後，y 的值為何？ (A) 15 (B) 16 (C) 20 (D) 8。
- (D) 8. 承上題，x 的值為何？ (A) 1 (B) 2 (C) 3 (D) 4。
- (B) 9. 下列哪一個 C++ 的運算式，可以取得 a 除 b 的餘數？
 (A) a/b (B) a%b (C) a mod b (D) a\b。
- (D) 10. 下列哪一個運算式執行後，k 的值與其它三者不同？
 (A) k++; (B) ++k; (C) k=k+1; (D) k+1;。
- (B) 11. 阿偉要在 C 語言程式中判斷兩個變數的值是否不相等，要使用下列哪一個關係運算子？ (A) <> (B) != (C) ~= (D) ><。
- (B) 12. 已知 A 為 true，B 為 false，下列哪一個運算式的結果為 true？
 (A) A && B (B) A && B || A (C) A && B || B (D) B || A && B。
- (B) 13. 下列何者不是 C 語言的邏輯運算子？ (A) && (B) ~ (C) || (D) !。
- (B) 14. 關於 C 語言的運算式，下列述敘何者錯誤？
 (A) 運算子「!」的優先權和「~」相同
 (B) 「=」是關係運算子，用來判斷左右兩邊的值是否相等
 (C) 執行 printf("%d", 2 && 5); 會輸出 1
 (D) 運算子「&」的優先權高於「|」。
- (C) 15. 變數 x 的值為 1，執行 printf("%i", ++x); 會輸出何值？
 (A) 0 (B) 1 (C) 2 (D) 3。

(B) 16. 變數 x 的值為 1，執行 `printf("%i", x++)`；會輸出何值？

(A)0 (B)1 (C)2 (D)3。

(D) 17. 以下程式碼片段執行後，變數 x 的值為何？

(A)0 (B)1 (C)2 (D)4。

```
int x = 0;
x = sizeof(float);
```

(B) 18. 若下列程式碼敘述為真，空格處應填入哪一種運算子？

(A)& (B)| (C)+ (D)^。

```
11 ___ 6 = 15;
```

(D) 19. 請問下列程式執行後的結果為何？

(A)2 (B)4 (C)8 (D)16。

```
int a=1;
cout << (a<<2<<2);
```

(A) 20. 請問下列程式執行後的結果為何？

(A)2 (B)4 (C)8 (D)16。

```
int a=32;
cout << (a>>2>>2);
```

答案 打★表示有詳解

1.(C) 2.(B) 3.(A) ★4.(C) ★5.(A) ★6.(C) ★7.(A) 8.(D) 9.(B) ★10.(D)
11.(B) ★12.(B) ★13.(B) ★14.(B) ★15.(C) ★16.(B) ★17.(D) ★18.(B) ★19.(D) ★20.(A)

解析

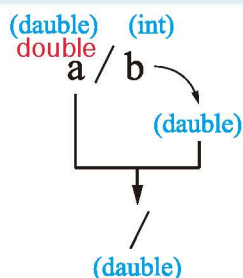
4. 5 不等於 4，結果為 `true(1)`。
5. 8 不小於 6，結果為 `false(0)`。
6. 運算式 `y=++x*5`；運算子 `++` 為前置運算，因此先將 x 加 1，再執行 `y=x*5`， $y=4*5=20$ 。
7. 運算式 `y=(x++)*5`；運算子 `++` 為後置運算，因此先執行 `y=x*5`，再將 x 加 1，故執行後 $y=15$ ， $x=4$ 。
10. (D)未使用指定運算子「`=`」將值指定給 k ， k 的內容不變。
12. 因為「`&&`」的優先權高於「`||`」，故選項(D)的運算順序為 `B||(A&&B)`。
13. 「`~`」是位元運算子。
14. (C)C 語言以「非 0」值表示 `true`，故條件式 `(2 && 5)` 為 `true`，回傳值為「1」。
15. 「`++x`」為前置運算，會先將 x 加 1，再輸出 x 。
16. 「`x++`」為後置運算，會先輸出 x ，再將 x 加 1。
17. `sizeof()` 會輸出資料型態所需的記憶體空間，因 `float` 資料型態的大小為 4Byte，故 `sizeof(float)` 會輸出 4。
18. 「`&`」、「`|`」、「`^`」為位元運算子，「`+`」為算術運算子，

(A) `11 & 6 = 2`
 (B) `11 | 6 = 15`
 (C) `11 + 6 = 17`
 (D) `11 ^ 6 = 13`
19. 「`<<`」為移位運算子，左移 n 位元，結果與數學運算乘上 2^n 相同，`a<<2<<2` = $1*2^2*2^2 = 16$ 。
20. 「`>>`」為移位運算子，右移 n 位元，結果與數學運算除以 2^n 相同，`a>>2>>2` = $32/2^2/2^2 = 2$ 。

3-2 資料型態轉換

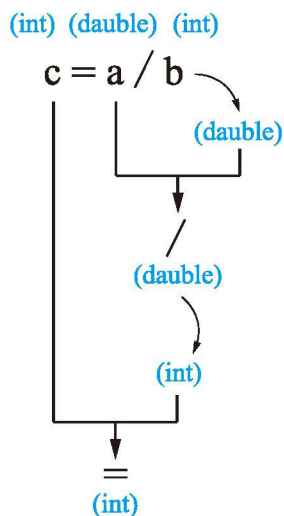
1. 當運算式中的運算元為不同資料型態的變數時，需進行資料型態轉換，資料型態轉換後，可能會造成資料不正確。
2. 自動轉換資料型態，由編譯器自動處理，
 - (1) 算術運算子的資料型態轉換：當運算元的資料型態不同時，資料型態較小的運算元，其值會先被轉成較大的資料型態，再進行運算，確保資料不會失真。

```
1. double a=5;          /* a 為 double 型態，大小為 8 Byte */
2. int b=2;             /* b 為 int 型態，大小為 4 Byte */
3.
4. printf("%f", a/b);    /* b 的值先轉成 double 型態(2.0)，再執行 a/b，結果為 2.5 */
```



- (2) 指定運算子「=」的資料型態轉換：「=」右邊的值會被轉成左邊變數的資料型態，當「=」左邊的資料型態較小時，可能會造成資料失真。

```
1. double a=5; /* a 為 double 型態，大小為 8 Byte */
2. int b=2;    /* b 為 int 型態，大小為 4 Byte */
3. int c=0;    /* c 為 int 型態，大小為 4 Byte */
4.
5. c=a/b;      /* 先進行 a/b，得到 double 型態的值 2.5，但指定給 c 時，2.5 會被轉成 int 型態，結果 c 的值為 2，產生資料失真 */
```



3. 在某些情況下，資料型態不會自動轉換，但會造成資料失真，

example

```
1. int a=5;          /* a 為 int 型態，大小為 4 Byte */
2. int b=2;          /* b 為 int 型態，大小為 4 Byte */
3.
4. printf("%f", a/b); /* 因為 a 和 b 都是 int 型態，因此 a/b 的結果將會是 2，而不是預期的 2.5 */
```

int / int --> int

4. 為避免上述情況造成資料失真，需使用**型態轉換運算子(cast type)**，將變數的值或運算式的結果強制轉換其資料型態，語法如下，

(資料型態)變數或運算式;

```
1. int a=5;          /* a 為 int 型態，大小為 4 Byte */
2. int b=2;          /* b 為 int 型態，大小為 4 Byte */
3.
4. printf("%f", (double)a/b); /* 將 a 強制轉換為 double 型態，b 則自動轉換為 double
                               型態，結果為 2.5 */
5. printf("%f", a/(double)b); /* 將 b 強制轉換為 double 型態，a 則自動轉換為 double
                               型態，結果為 2.5 */
6. printf("%f", (double)a/(double)b); /* 將 a 和 b 都強制轉換為 double 型態，結果為 2.5 */
7. printf("%f", (double)(a/b)); /* a/b 的結果為 2，再轉型為 double 型態，結果為 2.0 */
```

• 立即練習

- (A) 1. a、b、c、d 四個變數的資料型別分別為 double、float、int、char，則運算式「a*d-b/c」執行後的資料型態為何？ (A)double (B)float (C)int (D)char。
- (B) 2. 下列程式片段執行後，c 的值為何？ (A)2.5 (B)2 (C)2.0 (D)0。
 int a=5, b=2, c=0;
 c=a/b;
- (B) 3. 下列程式片段執行後，c 的值為何？ (A)2.5 (B)2 (C)2.0 (D)0。
 int a=5, b=2, c=0;
 c=(float)(a/b);
- (C) 4. 下列程式片段執行後，c 的值為何？ (A)2.5 (B)2 (C)2.0 (D)0。
 int a=5, b=2;
 float c=0;
 c=(float)(a/b);
- (A) 5. 下列程式片段執行後，c 的值為何？ (A)2.5 (B)2 (C)2.0 (D)0。
 int a=5, b=2;
 float c=0;
 c=(float)a/(float)b;