

程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH02 C程式設計入門



2.2 一個簡單的C程式

```
1  /* Fig. 2.1: fig02_01.c
2     A first program in C */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main( void )
7  {
8      printf( "Welcome to C!\n" );
9
10     return 0;
11
12 }
```

Welcome to C!

輸出控制

內縮
Tab

跳脫序列	說明
\n	換行。將游標移到下一行起始處。
\t	水平 tab。將游標移到下一個 tab 定位點。
\a	警告。讓系統發出警告聲。
\\	反斜線符號在字串中插入反斜線字元。
\"	雙引號。在字串中插入雙引號。

圖 2.2 一些常見的跳脫序列。

2.3 另一個簡單的C程式

```
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main( void )
7  {
8      int integer1;
9      int integer2;
10     int sum;
11
12     printf( "Enter first integer\n" );
13     scanf( "%d", &integer1 );
14
15     printf( "Enter second integer\n" );
16     scanf( "%d", &integer2 );
17
18     sum = integer1 + integer2;
19
20     printf( "Sum is %d\n", sum );
21
22     return 0;
23
24 }
```

Enter first integer
45
Enter second integer
72
Sum is 117

scanf("%d", &integer1);

輸入控制

練習

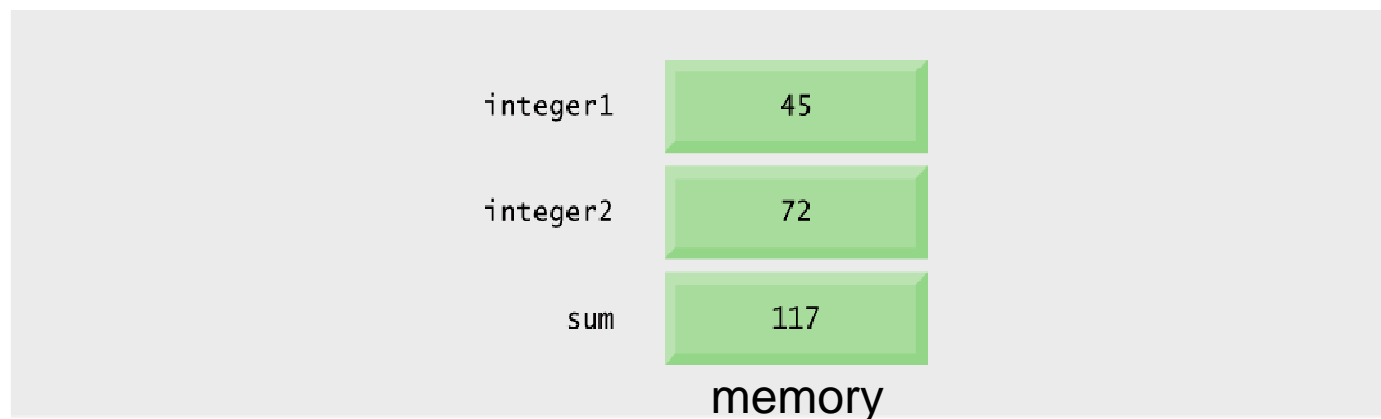
- 請為下列各敘述式找出錯誤並更正？
 - ❑ `scanf("d", value);`
 - ❑ `printf("The product of %d and %d is %d" \n, x, y);`
 - ❑ `scanf("%d", anInteger);`
 - ❑ `print("The sum is %d \n", x + y);`
 - ❑ `printf("The value you entered is %d \n", &value);`
- 請撰寫下列動作的C敘述式。
 - ❑ 顯示 **Enter two numbers** 訊息
 - ❑ 從鍵盤輸入三整數並將這些數值存在變數a, b, c中



2.4 記憶體觀念 (Memory)

■ 變數 (variables)

- ❑ 變數名稱會對應到電腦記憶體的某個位置
- ❑ 每個變數都具有名稱 (name)、型別 (type)、及數值 (value)
- ❑ 當數值放入變數時，會蓋掉(破壞)該位置原本的值
- ❑ 從記憶體中讀取變數時，並不會改變其中的值



變數的命名規則

■ 變數的組成

- 大寫字母、小寫字母、底線(_)、數字
- 名稱一開頭不可以是數字
- 大小寫不同
- 範例: SUM, sum, Sum_1, _fg, ... (合法)
3y, x\$2 (不合法)

■ 關鍵字, keywords (系統保留字, reserved words)

- 替C語言保留的特殊用字
- 不能做為識別字或變數名稱

關鍵字			
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

圖 2.15 C 關鍵字

變數的宣告

- 變數使用時，要先宣告變數並設定資料型態
 - 範例: `int i = 7, j, k;`
 - 溢位 (overflow)

		型態	長度 (bits)	值的範圍
數字	{	int	16	-32768~32767
		unsigned	16	0~65535
		long int	32	$-2^{31} \sim 2^{31}-1$
		unsigned long int	32	$0 \sim 2^{32}-1$
		float	32	
字元	{	double	64	
		char	8	0~255

2.5 C的算術運算

■ 算術計算

C 的運算	算術運算子 (arithmetic operator)	代數運算式	C 運算式
加法	+	$f + 7$	$f + 7$
減法	-	$p - c$	$p - c$
乘法	*	bm	$b * m$
除法	/	x / y or $\frac{x}{y}$ or $x \div y$	x / y
模數除法	%	$r \bmod s$	$r \% s$

■ 運算子優先順序

- 某些算術運算子會在其他運算子之前執行 (例如：乘法在加法之前)
- 範例: 計算變數a、b、c的平均值
 - 錯誤: $a + b + c / 3$
 - 正確: $(a + b + c) / 3$

運算子	運算	計算的順序 (優先順序)
()	小括號	優先計算。如果小括號是巢狀的，則會先計算最內層的。若有數個此類型的運算，則會從左到右運算。
*	乘法	第二個計算。若有數個此類型的運算，則會從左到右運算。
/	除法	
%	模數除法	
+	加法	最後計算。若有數個此類型的運算，則會從左到右運算。
-	減法	

圖2.10 算術運算子的優先順序

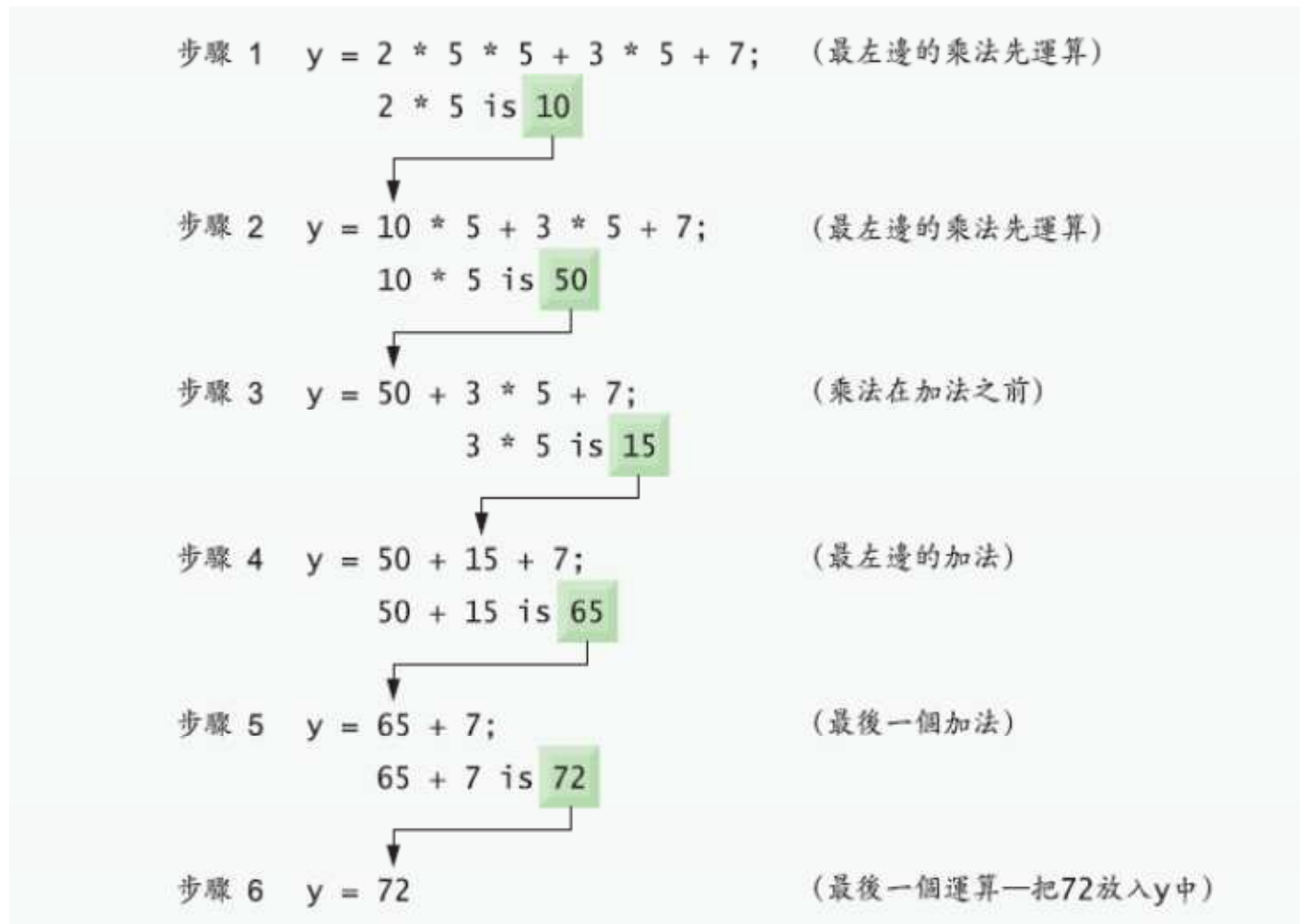


圖 2.11 二次多項式的計算順序

練習

- 若代數等式 $y=ax^3+7$ ，請問以下哪個C敘述可正確描述它？
 - ❑ a) $y = a * x * x * x + 7 ;$
 - ❑ b) $y = a * x * x * (x + 7) ;$
 - ❑ c) $y = (a * x) * x * (x + 7) ;$
 - ❑ d) $y = (a * x) * x * x + 7 ;$
 - ❑ e) $y = a * (x * x * x) + 7 ;$
 - ❑ f) $y = a * x * (x * x + 7) ;$

- 請說出下列C敘述的計算順序，並答出執行後的x值。
 - ❑ $x = 7 + 3 * 6 / 2 - 1 ;$
 - ❑ $x = 2 \% 2 + 2 * 2 - 2 / 2 ;$



2.6 判斷：等號運算子和關係運算子

■ 敘述式 (statement)

- 執行某一項動作 (計算、資料輸入／輸出)
- 進行某一項判斷
 - 依據考試成績的值印出「及格」或「不及格」

■ if 控制結構

- 若條件為真，則程式會執行if本體中的敘述式

```
...  
if (條件式)  
{  
    // 條件為真，執行if 區塊  
}  
...
```

標準代數的等號或關係運算子	C 的等號或關係運算子	C 的條件式範例	C 條件式的意義
等號運算子			
=	==	x == y	x 等於 y
≠	!=	x != y	x 不等於 y
關係運算子			
>	>	x > y	x 大於 y
<	<	x < y	x 小於 y
≥	>=	x >= y	x 大於或等於 y
≤	<=	x <= y	x 小於或等於 y

圖 2.12 等號運算子和關係運算子

```

4  #include <stdio.h>
7  int main( void )
8  {
9      int num1;
10     int num2;
11
12     printf( "Enter two integers, and I will tell you\n" );
13     printf( "the relationships they satisfy: " );
14     scanf( "%d%d", &num1, &num2 );
15
16     if ( num1 == num2 ) {
17         printf( "%d is equal to %d\n", num1, num2 );
18     }
19     if ( num1 != num2 ) {
20         printf( "%d is not equal to %d\n", num1, num2 );
21     }
22     if ( num1 < num2 ) {
23         printf( "%d is less than %d\n", num1, num2 );
24     }
25     if ( num1 > num2 ) {
26         printf( "%d is greater than %d\n", num1, num2 );
27     }
28     if ( num1 <= num2 ) {
29         printf( "%d is less than or equal to %d\n", num1, num2 );
30     }
31     if ( num1 >= num2 ) {
32         printf( "%d is greater than or equal to %d\n", num1, num2 );
33     }
34     return 0;
35 }

```

Enter two integers, and I will tell you
the relationships they satisfy: 3 7
3 is not equal to 7
3 is less than 7
3 is less than or equal to 7

練習

- 撰寫一個程式，要求使用者輸入兩個數字，再從使用者取得這兩個數字，然後印出這兩個數字的總和、乘積、差、商和餘數。
- 撰寫一個程式，要求使用者輸入兩個整數，然後印出比較大的那個整數以及 **is larger** 字串。假使兩個整數相等，則印出 **These numbers are equal.** 的訊息。