

6-1 公用函式

重點整理

1. **函式(Function)**或**副程式(Subroutine)**：將特定功能或常用的程式碼獨立成一個區塊，使用函式有以下優點：
 - (1) 程式模組化、結構化。
 - (2) 增加程式的可讀性。
 - (3) 使程式易於維護。
2. 主函式 `main()`也是一種函式，是程式執行的進入點。
3. C/C++語言提供標準函式庫，稱為公用函式或內建函式，一個函式庫就是多個同類型函式的集合。
4. **標頭檔(head file)**：副檔名為.h，用來描述函式庫內所有函式的特性和名稱，例如 `math.h` 是數學函式庫的標頭檔。
5. ANSI C 中常見的標準函式庫如下表：

函式庫	C 標頭檔	C++標頭檔
數學運算	<code><math.h></code>	<code><cmath></code>
輸入、輸出	<code><stdio.h></code>	<code><iostream></code>
日期、時間	<code><time.h></code>	<code><ctime></code>
數值轉換、亂數、動態記憶體	<code><stdlib.h></code>	<code><cstdlib></code>
字串處理	<code><string.h></code>	<code><cstring></code>

6. 在使用函式前，需先使用前置處理器指令引用函式庫，語法如下：

`#include <函式庫標頭檔>`

7. 常見的函式：

`<math.h>` 的數學函式

函式	說明	範例
<code>double exp(double x)</code>	指數 e^x	<code>exp(1) = 2.71828</code>
<code>double log(double x)</code>	自然對數 $\log_e x (\ln x)$	<code>log(2.71828) = 1</code>
<code>double log10(double x)</code>	10 為底的對數 $\log_{10} x$	<code>log10(100) = 2</code>
<code>double pow(double x, double y)</code>	x^y	<code>pow(10, 2) = 100</code>
<code>double sqrt(double x)</code>	x 的平方根	<code>sqrt(100) = 10</code>
<code>double ceil(double x)</code>	大於或等於 x 的最小整數	<code>ceil(5.4) = 6</code> <code>ceil(-5.4) = -5</code>
<code>double floor(double x)</code>	小於或等於 x 的最大整數	<code>floor(5.4) = 5</code> <code>floor(-5.4) = -6</code>
<code>double hypot(double x, double y)</code>	$\sqrt{x^2 + y^2}$	<code>hypot(6, 8) = 10</code>

函式	說明	範例
double round(double x)	四捨五入	round(4.4) = 4 round(4.5) = 5 round(-4.5) = -5
double abs(double x)	絕對值	abs(-3) = 3
double sin(double A)	正弦函式 sinA，參數 A 需以徑度表示	計算 $\sin 45^\circ$ ： 角度 45° ，則徑度 $= 45 \times 3.14 / 180 = 0.785$ $\sin(0.785) = 0.707$
double cos(double A)	餘弦函式 cosA，參數 A 需以徑度表示	計算 $\cos 45^\circ$ ： $\cos(0.785) = 0.707$
double tan(double A)	正切函式 tanA，參數 A 需以徑度表示	計算 $\tan 45^\circ$ ： $\tan(0.785) = 1$
double asin(double A)	反正弦函式 $\sin^{-1}A$ ，回傳徑度	$\sin(0.707) = 0.785$
double acos(double A)	反餘弦函式 $\cos^{-1}A$ ，回傳徑度	$\cos(0.707) = 0.785$
double atan(double A)	反正切函式 $\tan^{-1}A$ ，回傳徑度	$\tan(1) = 0.785$

<stdlib.h>的亂數函式

函式	說明	範例
void srand(unsigned int seed)	初始化亂數函式，指定亂數種子	srand(time(NULL)); time(NULL)為從 1970 年 1 月 1 日到目前為止的秒數，以此當做亂數種子
int rand(void)	隨機產生介於 0 到 32767 之間的整數	rand()%100; 產生 0~99 之間的亂數值 rand()%47; 產生 0~46 之間的亂數值 rand()%47+1; 產生 1~47 之間的亂數值

<string.h>的字串處理函式

函式	說明	範例
size_t strlen(char s[])	回傳字串長度，不包含結束字元'\0'	三個字元陣列如下： char s1[20] = "LeBron";
char[] strcpy(char s1[], char s2[])	複製 s2 到 s1	char s2[20] = "James";
char[] strncpy(char s1[], char s2[], size_t n)	複製 s2 的前 n 個字元到 s1	char s3[20] = "";
char[] strcat(char s1[], char s2[])	將 s2 連接到 s1 之後	strlen(s1); 取得 s1 的長度，回傳值為 6 (不包含結束字元'\0')
char[] strncat(char s1[], char s2[], size_t n)	將 s2 的前 n 個字元，連接到 s1 之後	strcpy(s3, s1); s3 的值為 LeBron strncpy(s3, s1, 2); s3 的值為 Le strcat(s1, s2); s1 的值為 LeBronJames strncat(s1, s2, 2); s1 的值為 LeBronJa strcmp(s1, s2); 比較字串 s1 和 s2，回傳值為 1(L 的 ASCII 碼為 76_{10} ，J 的 ASCII 碼為 74_{10} ，故 s1>s2)
int strcmp(char s1[], char s2[])	由左至右逐一比較 s1 和 s2 的每個字元(ASCII 碼)， s1=s2，回傳值=0 s1>s2，回傳值>0 s1<s2，回傳值<0	

註 以 `strcpy()` 為例，另一種表示方式是 `char *strcpy(char *s1, char *s2)`，其中 `char *s1` 和 `char[] s1` 都是表示「指向字元的指標」，但使用 `char[] s1` 可以更清楚的表示是「指向字元陣列的第 1 個元素」。

01 範例

設計一程式，由使用者輸入任意角度，計算其 `sin`、`cos`、`tan` 等三角函數後輸出。

解

程式碼：

```

1. #include <stdio.h>
2. #include <math.h>
3.
4. int main(){
5.     const double PI=3.14;
6.     double degree, radian;
7.     double sinVal, cosVal, tanVal;
8.
9.     printf("請輸入角度：");
10.    scanf("%lf", &degree);
11.
12.    radian = degree*PI/180;
13.    sinVal = sin(radian);
14.    cosVal = cos(radian);
15.    tanVal = tan(radian);
16.
17.    printf("sin%d = %lf\n", degree, sinVal);
18.    printf("cos%d = %lf\n", degree, cosVal);
19.    printf("tan%d = %lf\n", degree, tanVal);
20. }
```

程式碼說明：

行數	說明
2	引用 <code><math.h></code> 函式庫標頭檔。
5	宣告常數 <code>PI = 3.14</code> 。
9 ~ 10	使用者輸入角度。
12	計算徑度，徑度=角度* <code>3.14/180</code> 。
13 ~ 14	使用數學函式庫 <code><math.h></code> 提供的函式，計算三角函數。

02 範例

設計一剪刀、石頭、布的猜拳程式，由電腦隨機出拳，和使用者猜拳後輸出勝負結果。

解

程式碼：

```

1. #include <stdio.h>
2. #include <stdlib.h>
3. #include <time.h>
4.
5. int main(){
6.     int pc, user;
7.
8.     srand(time(NULL));
9.     pc=rand()%3;
10.
11.    printf("請輸入數字(0:剪刀 1:石頭 2:布)");
12.    scanf("%d", &user);
13.
14.    switch(pc){
15.        case 0:
16.            printf("電腦出剪刀,");
17.            if (user == 0) printf("你出剪刀，平手\n");
18.            else if (user == 1) printf("你出石頭，你贏了！\n");
19.            else if (user == 2) printf("你出布，你輸了！\n");
20.            break;
21.        case 1:
22.            printf("電腦出石頭\n");
23.            if (user == 0) printf("你出剪刀，你輸了\n");
24.            else if (user == 1) printf("你出石頭，平手！\n");
25.            else if (user == 2) printf("你出布，你贏了！\n");
26.            break;
27.        case 2:
28.            printf("電腦出布\n");
29.            if (user == 0) printf("你出剪刀，你贏了\n");
30.            else if (user == 1) printf("你出石頭，你輸了！\n");
31.            else if (user == 2) printf("你出布，平手！\n");
32.            break;
33.    }
34. }
```

程式碼說明：

行數	說明
1 ~ 3	引用函式庫標頭檔。
6	變數 pc 代表電腦出的拳，user 代表使用者出的拳。
8, 9	利用 srand()和 rand()函式，隨機產生 0 到 2 之間的整數，0、1、2 分別代表剪刀、石頭、布。
11, 12	使用者輸入數字，0、1、2 分別代表剪刀、石頭、布。
14 ~ 33	判斷勝負並輸出結果。

• 立即練習

- (C) 1. 關於 C 語言公用函式的描述，下列何者錯誤？
- (A)函式庫的標頭檔，副檔名為.h
 - (B)使用#include 指令，引用函式庫標頭檔
 - (C)函式一定有回傳值
 - (D)函式不一定要有參數。
- (A) 2. 程式敘述 rand()%3; 不可能會回傳下列何值？
- (A)3 (B)2 (C)1 (D)0。
- (A) 3. C 語言公用函式 sqrt(9)的回傳值為何？
- (A)3 (B)2 (C)1 (D)0。
- (D) 4. C 語言公用函式 pow(-4, 2)的回傳值為何？
- (A)2 (B)-2 (C)-16 (D)16。
- (C) 5. 佩佩要使用 C 語言的公用函式 sin()函式，計算 sin45°的結果，她必需先引用哪一個函式庫？ (A)string.h (B)stdlib.h (C)math.h (D)stdio.h。
- (D) 6. C 語言公用函式 floor(-3.7)的回傳值為何？
- (A)4 (B)3 (C)-3 (D)-4。
- (C) 7. C 語言公用函式 ceil(-3.7)的回傳值為何？
- (A)4 (B)3 (C)-3 (D)-4。
- (D) 8. 在 C 語言中執行 rand()%4; 指令，下列哪一個值不可能出現？
- (A)1 (B)2 (C)3 (D)4。
- (B) 9. 以下何者的值與其它 3 者不同？
- (A)abs(-5) (B)ceil(-5.3) (C)floor(5.3) (D)round(4.6)。
- (A) 10. 一個 RC 串聯電路，計算電容充電時的暫態電壓公式為 $V_C = E(1 - e^{-\frac{t}{RC}})$ ，要使用程式完成這個算式，下列哪一個函式最適合被使用？
- (A)exp() (B)log() (C)sqrt() (D)pow()。

(B) 11. 下列程式片段執行後的輸出為何？

- (A)9 (B)10 (C)11 (D)12。

```
int sum=0;
for (int i=-2;i<=2;i++){
    sum = sum + pow(i,2);
}
printf("%d", sum);
```

(B) 12. 在 C 語言中有兩個字串，str1[]="Orange"，str2[]="Apple"，則函式 strcmp(str1, str2) 的回傳值，最可能是下列何者？

- (A)0 (B)1 (C)-1 (D)皆不可能。

(B) 13. 承上題，執行 strcpy(str1, str2)後，str1 的內容為何？

- (A)AppleOrange (B)Apple (C)Orange (D)OrangeApple。

(B) 14. 接續上一題，若繼續執行 strcat(str1, str2)後，str1 的內容為何？

- (A)AppleOrange (B)AppleApple (C)OrangeOrange (D)OrangeApple。

(C) 15. 一函式 S()如下，S(3, 4)的回傳值為何？

- (A)24 (B)10 (C)25 (D)5。

```
double S(double x, double y){
    return pow(hypot(x, y),2);
}
```

(D) 16. 若函式 rand()的回傳值為一介於 0 和 10000 之間的亂數，下列那個運算式可產生介於 100 和 1000 之間的任意數(包含 100 和 1000)？

- | | |
|-------------------------|--------------------------|
| (A) rand() % 900 + 100 | (B) rand() % 1000 + 1 |
| (C) rand() % 899 + 101 | (D) rand() % 901 + 100。 |

(C) 17. 有兩個字串 s1 和 s2，分別為 "SuzanHola" 和 "AmigoFriend"，執行 strncat(s1, s2, 5)後，下列何者正確？

- | | |
|----------------------------|-----------------------------|
| (A)s1 = "AmigoFriendSuzan" | (B) s2 = "AmigoFriendSuzan" |
| (C)s1 = "SuzanHolaAmigo" | (D)s2 = "SuzanHolaAmigo"。 |

(A) 18. 在 C 語言中宣告 2 個字元陣列分別為 a[]="baseball"，b[]="basketball"，請問執行 strcmp(a, b); 後，回傳值為何？

- (A)-1 (B)0 (C)1 (D)2。

(D) 19. 關於 C/C++ 語言的公用函式，下列說明何者正確？

- (A)abs(double a)：回傳 a 的四捨五入值
- (B)floor(double a)：回傳 $\geq a$ 的最小整數
- (C)round(double a)：回傳 a 的絕對值
- (D)sqrt(double a)：回傳 \sqrt{a} 。

(C) 20. 指令 sqrt(abs(-49)); 執行後的輸出為何？

- (A)5 (B)6 (C)7 (D)9。

答案 打★表示有詳解

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

解析

9. (A)5 (B)-5 (C)5 (D)5。
10. 使用 `exp()`函式計算指數 e。
12. 比較第 1 個字元的 ASCII 碼，'O' 的 ASCII 碼為 79，'A' 的 ASCII 碼為 65，`str1` 大於 `str2`，回傳值 > 0。
17. 將 `s2` 的前 5 個字元，串接到 `s1`。
18. e 的 ASCII 碼為 101_{10} ，k 的 ASCII 碼為 107_{10} ，故 `s1` < `s2`，回傳 -1。

6-2 自訂函式**重點整理**

1. 函式是獨立的程式碼區塊，可以輸入外部資料，也可以將資料輸出。
2. 在程式中呼叫函式的時機：
 - (1) 在主函式 `main()` 中呼叫函式。
 - (2) 在函式 `A()` 中呼叫函式 `B()`。
 - (3) 在函式中呼叫自己，又稱為遞迴函式。
3. 引數與參數：
 - (1) 呼叫函式時，傳遞給函式的值，稱為 **引數(Argument)**。
 - (2) 函式被呼叫時，負責接收資料的變數，稱為 **參數(Parameter)**。
4. 當函式有回傳值時，在函式的程式碼區塊內，使用關鍵字 `return` 將資料回傳，回傳後函式隨即結束，而寫在 `return` 之後的程式碼，將不會被執行。
5. 若函式沒有回傳值，則在函式名稱之前使用 `void` 關鍵字表示沒有回傳值。
6. 自訂函式包含宣告和定義(實作)2 個部份，語法如下：

- (1) **宣告**函式原型：規範回傳值與參數的資料型態，需在主函式 `main()` 與函式定義之前宣告。

回傳值資料型態 函式名稱(參數資料型態, ...);

```

1. void A();           /* 宣告函式原型 A()，沒有參數，也沒有回傳值 */
2. void B(int, int);  /* 宣告函式原型 B()，2 個整數型態的參數，沒有回傳值 */
3. int C(float);     /* 宣告函式原型 C()，1 個浮點數型態的參數，回傳整數 */

```

- (2) **定義函式**：負責參數宣告與程式區塊實作，若定義寫在主函式 main()之前，則可以省略上述的函式原型宣告。

回傳值資料型態 函式名稱(參數資料型態 參數名稱, ...){ 程式碼區塊 }

```
4. void A(){ ... }
5. void B(int a, int b){ ... }      /* 宣告 2 個整數參數 a 和 b，同時實作程式碼區塊 */
6. int C(float x){ ... }           /* 宣告 1 個浮點數參數 x，同時實作程式碼區塊 */
```

7. 呼叫函式時，會暫停目前執行的程式碼，待函式執行結束後，再返回原來的程式碼位置繼續執行，呼叫函式的語法如下(引數的數量與型態，需符合函式原型的規範)，

函式名稱(引數, ...);

```
7. A();                  /* 呼叫函式 A()，沒有引數 */
8. B(10, 20);            /* 呼叫函式 B()，引數是整數 10 和 20 */
9. int i=C(3.14);        /* 呼叫函式 C()，引數是浮點數 3.14，回傳值指定給整數變數 i */
```

8. 以下範例說明函式呼叫的過程，

```
int C(float);    /* 宣告函式原型C() */
int C(float x){ /* 定義函式C() */
    printf("%f",x);
    return x*100;
    printf("return之後不會執行");
}
int main(void){
    int i = C(3.14);
    printf("%d",i);
}
```

- (1) 從主函式 main()開始執行。
- (2) 呼叫函式 C(3.14)，傳入的引數為 3.14，函式 C()被呼叫後，會宣告一參數 x 並指定為 3.14(float x = 3.14)。
- (3) 執行函式 C()，直到遇到 return 指令，回傳 x*100，函式執行結束，返回原程式碼繼續執行。
- (4) 將函式回傳值指定給變數 i(int i=314)。
- (5) 寫在 return 之後的程式碼，不會執行。

9. 區域變數與全域變數

	區域變數	全域變數
種類	- if、for、while 結構內宣告的變數 - 函式的參數 - 函式內部宣告的變數	函式外部宣告的變數
存取權	只能在所在的程式區塊(即左右大括號{}包圍的部份)內被存取	在程式的任何位置都可以存取
生命週期	程式區塊結束即消失	程式結束才消失

10. 程式執行時，每個變數存在的時間不同，稱為變數的**生命週期**。下列程式碼宣告 1 個全域變數 a 和 3 個區域變數 b、c 和 i，變數建立(作業系統配置記憶體)和消失(作業系統回收記憶體)的時間順序如下，其中 for 迴圈內的區域變數 c，每執行一次迴圈即消失(5)，進入迴圈時再重新建立(4)，共重覆 10 回合。

- (1) 建立全域變數 a
- (2) 建立 main() 的區域變數 b
- (3) 建立 A() 的區域變數 i
- (4) 建立 for 迴圈的區域變數 c
- (5) for 迴圈的區域變數 c 消失
- (6) A() 的區域變數 i 消失
- (7) main() 的區域變數 b 消失
- (8) 全域變數 a 消失

```

1. int a=1;           /* 全域變數 a */ ----- (1)
2.
3. void A(){
4.     int i=0;           /* 函式 A() 的區域變數 i */ ----- (3)
5.     for(i=0; i<10; i++){
6.         int c=0;           /* for 迴圈的區域變數 c */ ----- (4)
7.         c=c+i;
8.     } ----- (5)
9. } ----- (6)
10.
11. int main(void)
12. {
13.     int b=0;           /* 主函式 main() 的區域變數 b */ ----- (2)
14.     A();
15. } ----- (7) (8)

```

11. 不同程式區塊的區域變數，因存在於記憶體的不同位置，因此允許使用相同的變數名稱，存取時，採由內而外的方式，優先存取同一程式區塊內的區域變數，若區域變數不存在，則往上一層尋找。下例的第 1、4、6、18 行皆宣告一名稱為 s 的變數，虛線指向存取的對象，第 13 行因函式 B() 內無區域變數 s，故存取對象為上一層的全域變數 s。

```

1. int s = 2; /* 全域變數 s */
2.
3. void A(){
4.     int s = 1; /* 函式 A() 的區域變數 s */
5.     for (int i=0;i<10;i++){
6.         int s = 0; /* for 迴圈結構的區域變數 s */
7.         s=s+1;
8.     }
9.     s=s+1;
10. }
11. void B(int a, int b)
12. {
13.     s=s+1;
14. }
15.
16. int main(void)
17. {
18.     int s = 3; /* 主函式 main() 的區域變數 s */
19.     A();
20.     B();
21.     s=s+1;
22. }
```

12. **重載(Overload)**函式：C++允許在函式的參數個數或參數資料型態不同時，使用相同的函式名稱，下例為 3 個名稱相同的重載函式，根據參數的不同，自動呼叫不同的函式。

```
void myFun(int, int, int) { ... }
```

```
void myFun(int) { ... }
```

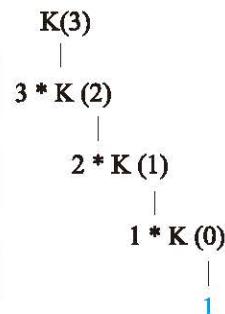
```
void myFun(double) { ... }
```

註 C 語言不允許函式重載，但允許運算子重載，如運算子「-」可以是負號或減號兩種功能。

13. 函式自己呼叫自己，稱為遞迴函式(Recursive Function)，遞迴函式必須具備**至少一個終止條件**，否則函式將無法結束，下面是一個計算 $n!$ (n 階層)的函式 $K()$ ，執行 $K(3)$ 會回傳 $3 \times 2 \times 1 \times 1$ ，函式執行的過程如下。

```

1. int K(int n){
2.     if (n == 0)      /* 終止條件：當 n=0 時，不再呼叫自己 */
3.         return 1;
4.     else
5.         return n * K(n-1); /* 呼叫自己 */
6. }
```



14. 遞迴函式的優點是程式較簡潔明確且可讀性較高，缺點是執行效率低，會佔用較多記憶體空間。

03 範例

來練題

世界衛生組織建議以身體質量指數(BMI)來衡量肥胖程度，請建立一自訂函式 $bmi()$ ，該函式會回傳 BMI 值。BMI 的計算公式為 $BMI = \frac{\text{體重(kg)}}{\text{身高(m)}^2}$

解

程式碼：

```

1. #include <stdio.h>
2. #include <math.h>
3.
4. double bmi(double, double); /* 廣告函式原型 */
5.
6. double bmi(double height, double weight){ /* 函式定義 */
7.     double bmiVal;
8.     bmiVal=weight/pow(height/100, 2);
9.     return bmiVal;
10. }
11.
12. int main(){
13.     double height, weight;
14.
15.     printf("請輸入身高(cm):");
16.     scanf("%lf", &height);
17.     printf("請輸入體重(kg):");
18.     scanf("%lf", &weight);
19.
20.     printf("%lf", bmi(height, weight)); /* 呼叫函式 bmi()，直接輸出回傳值 */
21. }
```

程式碼說明：

行數	說明
2	引用數學函式庫<math.h>。
4	宣告 bmi()函式原型：規範 2 個 double 型態的參數，回傳值也必需是 double 型態。
6 ~ 10	函式定義：實作 BMI 的計算公式，參數 height 和 weight 負責接收傳入的引數。
8	計算 BMI 值，利用 pow()函式進行次方運算。
9	回傳 BMI 值。
15 ~ 18	使用者輸入身高和體重。
20	呼叫 bmi()函式，提供引數 height 和 weight。
6, 13	bmi()和 main()都有 height 和 weight 變數，因為不同函式的區域變數彼此獨立，所以變數名稱相同並不會造成編譯錯誤。

04 範例

練習題

費式數列，由義大利人費先生提出，該數列由 0 和 1 開始，之後的每個數為前兩個數字相加，舉列如下：0, 1, 1, 2, 3, 5, 8, 13, 21, 34...，現在，費先生請你幫忙設計一遞迴函式 fib()，該函式可以直接回傳一費式數列的值(例如 fib(5)會回傳 5)。

解

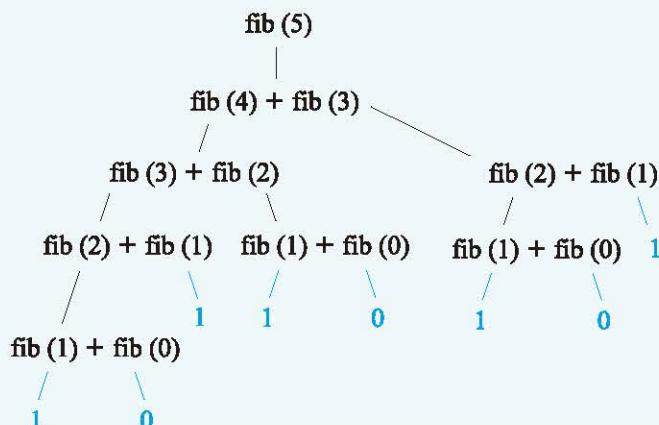
程式碼：

```

1. int fib(int);
2.
3. int fib(int x){
4.     if (x == 0){      /* 終止條件一 */
5.         return 0;
6.     }
7.     else if (x == 1){ /* 終止條件二 */
8.         return 1;
9.     }
10.    else{
11.        return fib(x-1)+fib(x-2); /* 呼叫自己 */
12.    }
13. }
```

程式碼說明：

行數	說明
1	函式原型宣告，規範 1 個 int 型態的參數，且需回傳一 int 資料。
3 ~ 13	fib()函式定義，有兩個終止條件，當 $x==0$ 或 $x==1$ 時，函式結束，並回傳 0 或 1。
11	不符合終止條件時，fib()會呼叫自己，以 fib(5)為例，此遞迴函式的執行過程如下，



05 範例

素數題

就讀國小六年級的表妹正在學習利用輾轉相除法來取得 2 個整數的最大公因數，請你幫表妹設計一遞迴函式 GCD()，該函式會回傳 2 個整數的最大公因數(例如 GCD(9, 12)會回傳 3)。

解

程式碼：

```

1. int GCD(int, int);
2.
3. int GCD(int x, int y){
4.     int r;
5.     r = x % y;
6.     if (r == 0)           /* 終止條件 */
7.         return y;
8.     return GCD(y, r);   /* 呼叫自己 */
9. }
```

程式碼說明：

行數	說明
1	函式原型宣告，規範 2 個整數型態的參數，且需回傳一整數資料。
3 ~ 9	GCD()函式定義，終止條件為 $r==0$ ，此時函式結束並回傳最大公因數 y。
11	不符合終止條件時，GCD()會呼叫自己，以 GCD(9, 12)為例，此遞迴函式的執行過程如下， $\begin{array}{c} \text{GCD}(9, 12) \\ \downarrow r=9 \\ \text{GCD}(12, 9) \\ \downarrow r=3 \\ \text{GCD}(9, 3) \\ \downarrow r=0 \text{ (終止條件)} \\ 3 \end{array}$

• 立即練習

- (D) 1. 關於函式的描述，下列何者正確？

(A)函式一定要有回傳值 (B)遞迴函式只能有一個終止條件
(C)自訂函式一定要宣告函式原型 (D)函式可以傳入多個參數。

(B) 2. 要從函式將值回傳，要使用哪一個關鍵字？ (A)back (B)return (C)tx (D)send。

(C) 3. 要建立自訂函式時，若該函式不需要回傳值，則在宣告函式原型時，回傳值的資料型態為何？ (A)null (B)nothing (C)void (D)no。

(D) 4. 下列程式碼中，哪一個變數是全域變數？ (A)m (B)n (C)k (D)sum。

```
int sum=0;
void F(int n){
    int m=n;
    sum=sum*m;
}
int main(){
    int k=10;
    F(k);
}
```

(D) 5. 關於 C++ 語言的函式，下列哪一個說法是錯誤的？

(A)函式自己呼叫自己，稱為遞迴函式
(B)在函式內部宣告的變數，稱為區域變數
(C)全域變數與區域變數的名稱可以相同
(D)在 main() 中呼叫函式時所傳的值，稱之為參數。

(B) 6. 下列有關 C 語言的敘述，哪一個是錯誤的？

- (A)同一陣列的所有元素，一定是相同的資料型別 (B)函式一定要有回傳值
(C)使用變數前未宣告，會產生編譯錯誤 (D)使用編譯程式轉換成機器語言。

(C) 7. 小豪宣告一個宣式原型 int PI(double, double)，該函式不可能回傳哪一個數值？

素養題 (A)0 (B)-1 (C)3.14 (D)1。

(B) 8. 下列 C 程式執行後的輸出為何？ (A)0 (B)1 (C)2 (D)3。

```
int A(int a){  
    a--;  
    return a;  
}  
void B(int b){  
    b++;  
    printf("%i", A(b));  
}  
int main(void)  
{  
    int x=1;  
    B(x);  
}
```

(A) 9. 執行下列程式片段後的輸出結果為何？ (A)5 0 (B)5 5 (C)0 0 (D)5 5。

```
int x=0;  
void Fun(int a){  
    int x=0;  
    x=x+a;  
    printf("%i", x);  
}  
int main(void)  
{  
    Fun(5);  
    printf("%i", x);  
}
```

(B) 10. 執行下列程式片段後的輸出結果為何？ (A)0 (B)1 (C)10 (D)11。

```
int s = 1;
void Y(){
    int s = 1;
    for (int i=0;i<10;i++){
        s=s+1;
    }
}
int main(void)
{
    Y();
    printf("%i", s);
}
```

(B) 11. 下列程式的輸出結果為何？ (A)1024 (B)512 (C)256 (D)2048。

```
#include <stdio.h>
int a=128;
int X(int b)
{
    int i;
    for( i=0;i<4;i++)
        b*=2;
    return (b);
}
int main()
{
    int a=32;
    a=X(a);
    printf("%d\n", a);
}
```

(C) 12. 一程式片段如下，F(7)的結果為何？ (A)10 (B)11 (C)12 (D)13。

```
int F(int x){
    if (x<0)
        return 1;
    else
        return F(x-2) + F(x-3);
}
```

(B) 13. 下列程式執行後的輸出為何？ (A)55 (B)50 (C)0 (D)10。

```
int s=50;
void K(int a){
    int s=0;
    for(int i=1;i<=a;i++){
        s=s+i;
    }
}
int main(){
    int a=10;
    K(a);
    printf("%d", s);
}
```

(B) 14. 下列函式兩個回傳式分別該如何撰寫，才能正確計算並回傳兩參數 a, b 之最大公因數(Greatest Common Divisor)？

- (A)a, GCD(b,r) (B)b, GCD(b,r) (C)a, GCD(a,r) (D)b, GCD(a,r)。

```
int GCD (int a, int b) {
    int r;
    r = a % b;
    if (r == 0)
        return _____;
    return _____;
}
```

(B) 15. 下列給定下列程式，其中 s 有被宣告為全域變數，請問程式執行後輸出為何？

- (A) 1,6,7,7,8,8,9 (B) 1,6,7,7,8,1,9 (C) 1,6,7,8,9,9,9 (D) 1,6,7,7,8,9,9。

```
int s = 1; // 全域變數
void add (int a) {
    int s = 6;
    for( ; a>=0; a=a-1) {
        printf("%d,", s);
        s++;
        printf("%d,", s);
    }
}
int main() {
    printf("%d,", s);
    add(s);
    printf("%d,", s);
    s = 9;
    printf("%d,", s);
    return 0;
}
```

(D) 16. 下列程式關於 C 語言函式的敘述，何者正確？

- (A)hypot()為自訂函式
- (B)呼叫自訂函式時，需在函式前面加上 call 關鍵字
- (C)函式若有傳回值，則需在定義時，在函式的前面加上 void 關鍵字
- (D)在自訂函式中，可以呼叫自己本身這個函式。

答案 打★表示有詳解

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

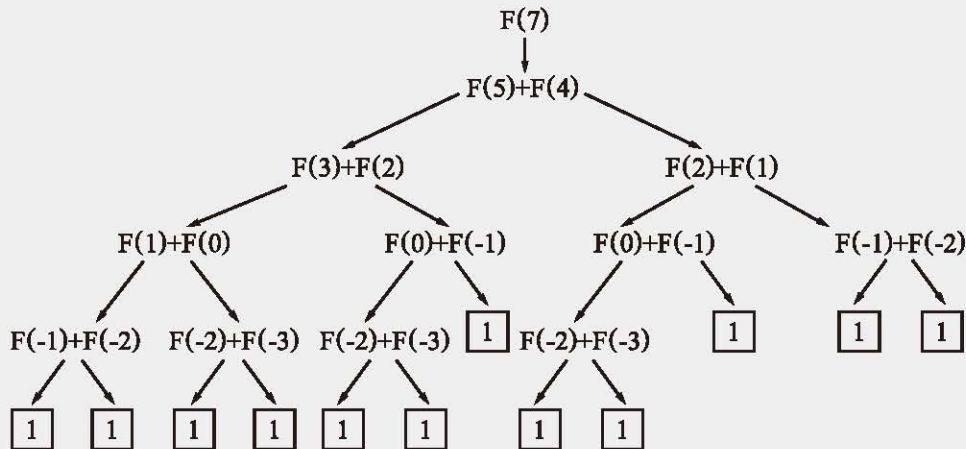
解析

1. (A)函式不一定要有回傳值 (B)可以有多個終止條件 (C)不一定要宣告函式原型。
5. (D)呼叫函式時所傳入的值，稱為引數。
7. 回傳值需為整數。
9. 區域變數與全域變數存在於記憶體的不同位置，可視為獨立的變數，允許使用相同名稱。此程式在函式 Fun() 內對其區域變數 x 進行運算，並不會影響全域變數 x。
10. 參考第 9 題。
11. 程式執行步驟如下，

```
#include<stdio.h>
int a=128;
int X(int b)
{
    int i;
    for(i=0;i<4;i++)
        b*=2;
    return(b);
}
int main()
{
    int a=32;
    a=X(a);
    printf("%d\n",a);
}
```

-
- (1) 呼叫 X(a)，引數為區域變數 a 的值，函式 X() 被呼叫後，參數 b 的值等於 32，
 - (2) 運算後，b 等於 512，
 - (3) 函式 X() 結束，返回原程式位置，回傳 512，並將回傳值指定給區域變數 a。

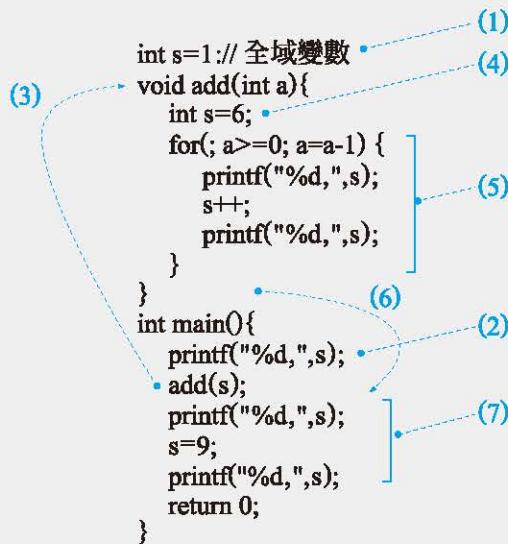
12. 呼叫 $F(7)$ 的執行過程如下，



13. 參考第 9 題解析。

14. 參考單元 6-2 範例 5。

15. 程式執行步驟如下，



- (1) 宣告全域變數 s，初始值為 1，
- (2) 輸出全域變數 s，
- (3) 呼叫 add(s)，傳入的引數為全域變數 s 的值，函式被呼叫後，參數 a 等於 1，
- (4) 宣告區域變數 s，此變數只在 add()函式內有效，函式結束即消失，因為區域變數與全域變數的存取範圍不同，故允許同名，
- (5) 輸出區域變數 s 的值：6 7 7 8，
- (6) 函式 add()結束，返回原程式位置，
- (7) 輸出全域變數 s 的值：1 9。

16. (A)hypot()為公用函式

- (B)呼叫自訂函式時，不需在函式之前加上任何關鍵字
- (C)函式若有傳回值，則需在定義時，在函式的前面加上傳回值的資料型態
- (D)函式可以呼叫自己本身，稱為遞迴函式

6-3 函式的資料傳遞

重點整理

- 函式使用**參數(parameter)**來接收外部傳入的資料，資料可以是變數或陣列。
- 傳遞變數的方式：

(1) 傳值呼叫(call by value)

資料以**值**的型式傳遞，函式的參數是**一般變數**，其值等於**來源變數的值**，參數和來源變數分別存在於記憶體的不同位置，所以**參數的改變不會影響來源變數的值**。

以下列程式碼為例，在 `byValue()` 內部將參數 `x` 指定為 100，並不會改變 `main()` 中變數 `a` 的值(`a` 的值仍然是 10)。

```

1. void byValue(int x){ /* 參數 x 是一般變數 */
2.     printf("%d\n", x); /* 輸出 10 */
3.     x = 100;           /* 指定參數 x 的值為 100 */
4. }
5. int main(){
6.     int a = 10;
7.     byValue(a);        /* 呼叫 byValue()，傳遞的引數是變數 a 的值 */
8.     printf("%d\n", a); /* 變數 a 的值不變，依舊是 10 */
9. }
```

程式執行時，變數在記憶體內的變化如下：

第 6 行，宣告變數 `a`

a			
10			

第 7 行，呼叫函式 `byValue(a)`，採傳值呼叫，等同於執行 `int x=a;`

a	x		
10	10		

第 3 行，將參數 `x` 指定為 100

a	x		
10	100		

第 4 行，函式結束，參數 `x` 消失，返回第 8 行

a	x		
10	100		

(2) 傳址呼叫(call by address)

資料以**位址**的型式傳遞，函式的參數是**指標變數**，其值是**來源變數的位址**，當**參數指向的記憶體內容改變時，來源變數的值也會跟著改變**。

以下列程式碼為例，第 7 行呼叫 `byAddr()` 時，傳入的引數是變數 `a` 的位址，此時 `byAddr()` 的參數 `x` 會指向來源變數 `a`，所以當 `x` 指向的記憶體內容被指定為 100 時，`main()` 中變數 `a` 的值也會改變(`a` 的值變成 100)。

```

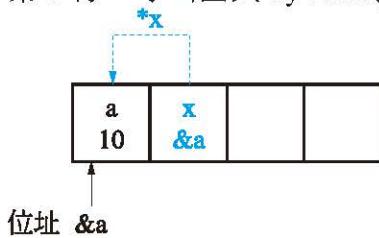
1. void byAddr(int *x){      /* 參數 x 是指標變數 */
2.     printf("%d\n", *x);   /* 輸出 10 */
3.     *x = 100;            /* 指定參數 x 指向的記憶體位址的內容為 100 */
4. }
5. int main(){
6.     int a = 10;
7.     byAddr(&a);         /* 呼叫 byAddr()，傳遞的引數是變數 a 的位址 */
8.     printf("%d\n", a);   /* 變數 a 的值已經變成 100 */
9. }
```

程式執行時，變數在記憶體內的變化如下：

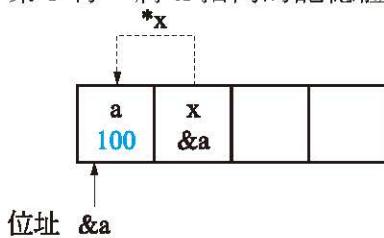
第 6 行，宣告變數 a



第 7 行，呼叫函式 byValue(&a)，採傳址呼叫，等同於執行 `int *x=&a;`



第 3 行，將 x 指向的記憶體內容，指定為 100



第 4 行，函式結束，參數 x 消失，返回第 8 行



註 取址運算子「&」，取得某個變數的記憶體位址。

取值運算子「*」，存取指標變數指向的記憶體內容。

3. 傳遞陣列資料時，需同時提供陣列和元素個數，函式有 2 種接收陣列資料的方式：

(1) 以陣列的方式接收，在函式內部使用陣列索引存取陣列元素

```

1. /* 宣告函式原型：規範參數的型態為整數陣列(用 int[] 表示)和整數變數 */
2. void G(int[], int);
3.
4. /* 定義函式：宣告 2 個參數分別為整數陣列 x[] 和整數變數 xLen */
5. void G(int x[], int xLen){
6.     for (int i=0; i<xLen; i++){
7.         x[i]=x[i]+1; /* 以陣列索引的方式存取陣列元素 */
8.     }
9. }
```

(2) 以指標的方式接收，在函式內部使用指標存取陣列元素

```

1. /* 宣告函式原型：規範參數的型態為指標變數和整數變數 */
2. void F(int*, int);
3.
4. /* 函式定義：宣告 2 個參數分別為指標變數 x 和整數變數 xLen */
5. void F(int *x, int xLen){
6.     for (int i=0; i<xLen; i++){
7.         *(x+i) = *(x+i)+1; /* 以指標的方式存取陣列元素 */
8.     }
9. }
```

4. 呼叫上述函式 G() 或函式 F()，均需要提供 2 個引數，分別為**陣列名稱(第 1 個元素的指標)**和**元素個數**，只要在函式內改變陣列的內容，來源陣列的內容也會跟着改變。

```

1. int main(){
2.     int a[5] = {1,2,3,4,5};
3.     G(a, 5);           /* 呼叫函式 G()，函式結束後，陣列內容變成 2 3 4 5 6 */
4.     F(a, 5);           /* 呼叫函式 F()，函式結束後，陣列內容變成 3 4 5 6 7 */
5. }
```

06 範例

素養題

某量販店推出消費打 9 折，滿千再送百的快閃活動，即消費金額打 9 折，折後每滿 1000 元，再贈送 1 張 100 元的折價卷，請以傳址呼叫的方式設計一函式 shopback()，計算打折後的消費金額，並顯示獲得的折價卷張數。

解

程式碼：

```

1. #include <stdio.h>
2.
3. void shopback(int*); /* 函式原型宣告，規範參數是指標變數 */
4.
5. void shopback(int *x){ /* 告知參數 x 是指標變數 */
6.     int coupon;
7.
8.     *x = (*x)*0.9;
9.     coupon = *x / 1000;
10.
11.    printf("獲得 %d 張折價卷\n", coupon);
12. }
13.
14. int main(){
15.     int m;
16.
17.     printf("請輸入消費金額：");
18.     scanf("%d", &m);
19.
20.     shopback(&m); /* 呼叫 shopback()，引數是變數 m 的位址 */
21.     printf("打折後的消費金額 %d", m);
22. }
```

程式碼說明：

行數	說明
3	shopback()函式原型宣告，規範參數的型態為指標變數。
5 ~ 12	shopback()函式定義，當第 20 行呼叫 shopback()時，參數 x 會指向 main()中的變數 m。
8	使用取值運算子「*」，存取 x 指向的記憶體內容，當*x 變更時，main()中的變數 m 也會跟著改變。
20	呼叫 shopback()函式，引數是變數 m 的位址（使用取址運算子「&」）。

07 範例

設計一自訂函式 arrMax()，傳入整數陣列，該函式會回傳陣列中最大值的索引，並使用此函式改寫 P5-5 範例 03。

解

程式碼：

```

1. #include <stdio.h>
2.
3. int arrMax(int[], int);
4.
5. int arrMax(int arr[], int len){
6.     int max=1;
7.     for (int i=2;i<=len;i++){
8.         if (arr[i-1]>arr[max-1])
9.             max = i;
10.    }
11.    return max;
12. }
13.
14. int main(){
15.     int money[] = {900,850,925,1012,936,998,1019,899,986,978,1002,1011};
16.     int max = arrMax(money, 12);
17.     printf("營業額最高的月份 : %d 月\n", max);
18.     printf("營業額 : %d 仟萬元", money[max-1]);
19. }
```

程式碼說明：

行數	說明
3	arrMax()函式原型宣告，規範參數的型態為整數陣列和整數變數。
5 ~ 12	arrMax()函式定義，宣告 2 個參數，分別為整數陣列 arr[] 和整數變數 len，使用陣列索引的方式存取陣列，找出最高營業額的月份 max，將 max 回傳。
16	呼叫 arrMax()函式，提供 2 個引數，分別為陣列名稱和元素個數。

08 範例

寫出以下程式執行後之輸出結果？

```

1. void F(char a[]) {
2.     int n, i;
3.     char t;
4.     if (a[0] != '\0') {
5.         for (n=0; a[n]!='\0'; n++);
6.         t = a[n-1];
7.         for (i=n-1; i>0; i--) {
8.             a[i] = a[i-1];
9.         }
10.        a[0] = t;
11.        F(a+1);
12.    }
13. }
14. int main() {
15.     char a[] = "HELLOWORLD";
16.     F(a);
17.     printf("%s", a);
18.     return 0;
19. }
```

解

行數	說明
4	終止條件，當傳入陣列 a 的第 1 個元素 a[0]為結束字元'\0'時，條件式不成立，函式 F()結束。
5	使變數 n 為傳入陣列 a 的大小（不包含結束字元'\0'）。
6	使變數 t 為傳入陣列 a 的最後一個字元。
7 ~ 9	將傳入陣列 a 的每一個元素都往後移一格（不包含結束字元'\0'）。
10	指定陣列 a 的第 1 個元素為 t（即將最後一個字元移到最前面）。
11	F()呼叫自己，引數為傳入陣列 a 的第 2 個元素的位址。

行數	說明	
16	遞迴函式 F()的執行過程如下，最後字元陣列 a[]的內容為 DLROWOLLEH。	
	呼叫 F()的次數	傳入的陣列內容
1	a ↓ H E L L O W O R L D \0	a ↓ D H E L L O W O R L \0
2	a+1 ↓ D H E L L O W O R L \0	a+1 ↓ D L H E L L O W O R \0
3	a+2 ↓ D L H E L L O W O R \0	a+2 ↓ D L R H E L L O W O \0
4	後續步驟請讀者自行練習	

• 立即練習

- (C) 1. 今有一函式原型 void K(int*)和整數變數 data，下列何者是呼叫函式 K()的正確語法？
 (A)K(data) (B)int x=K(data) (C)K(&data) (D)int x=K(&data)。
- (D) 2. 函式呼叫有兩種方式：傳值呼叫(call by value)和傳址呼叫(call by address)，下列何者不正確？
 (A)傳址呼叫，不能將函式中宣告的變數位址回傳
 (B)傳值呼叫，參數在函式的變化不會影響本來的變數
 (C)傳址呼叫，參數在函式內的變化會直接改變本來的變數
 (D)傳值呼叫，參數必需是指標變數。
- (A) 3. 執行函式呼叫時，若傳入的來源變數不希望被改變，可採用何種呼叫方式？
 (A)傳值呼叫 (B)傳址呼叫 (C)傳名呼叫 (D)遞迴呼叫。
- (A) 4. 下列程式碼執行後的輸出為何？ (A)10,20 (B)20,10 (C)10,10 (D)20,20。

```
void swap(int a, int b){
    int tmp;
    tmp=a;
    a=b;
    b=tmp;
}
int main(){
    int x=10,y=20;
    swap(x,y);
    printf("%d,%d",x,y);
}
```