

Chapter9 函式

一、為何我們需要函式？

1. 使程式容易管理
2. 軟體再使用 (abstraction)
3. 避免重複撰寫同一程式碼

二、函式定義的格式

格式	例子
回傳值型態 函式名稱(參數列) { 變數宣告; 陳述句; return 回傳值; }	<pre>int sum(int a, int b) { int n; n = a + b; return n; }</pre>

三、函式原型

編譯器會利用函式原形去驗證函式的呼叫，函式原形通常置於 main 函式外，或是標頭檔內。函式原形的另一個功能是引數型態的強制轉換。

格式	例子
傳回值型態 函式名稱 (引數型態);	<pre>int sum(int, int); 或 int sum(int a, int b);</pre>

標頭檔 (Header File, .h) 裡面含有函式原形，以及這些函式所需的各種「資料型態」和「常數」的定義。我們可以撰寫自己的標頭檔，以.h 為附檔名。在原始程式檔中，以#include 將標頭檔囊括進去。

四、函式的呼叫

1. 傳值呼叫 (call by value)：將引數的「值」拷貝一份傳給函式，不會影響原來的變數值。
2. 傳參考呼叫 (call by reference)：將引數的「位址」傳給函式，會影響原來的變數值。

五、遞迴的函式

1. 自己呼叫自己，注意終止條件的設定
2. 用到大量的堆疊 (stack) 空間，容易造成記憶體不足
3. 可以改寫成迴圈形式

4. 舉例說明：

(1) 河內塔 (Hanoi)：

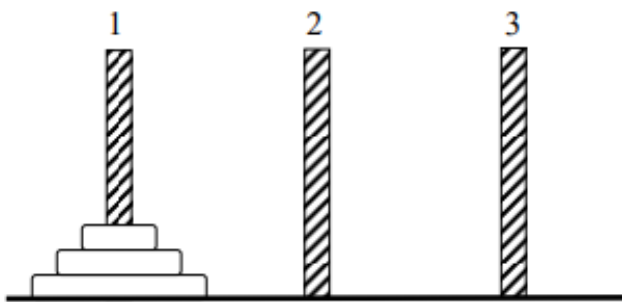
I. 說明：

有三根柱子和 n 個大小不同的環 (如圖)。

欲將 n 個環由 A 柱移到 B 柱上，移動規則如下：

1. 一次只能移動一個環
2. 在搬移的過程中，大環不能壓住小環。

試問：最少移動次數及搬移過程為何？



II. 程式碼：

```
void hanoi(int n, int src, int dst, int tmp)
{
    if (n==1)
        printf("%d → %d\n", src, dst);
    else
    {
        hanoi(n-1, src, tmp, dst);
        printf("%d ( %d\n", src, dst);
        hanoi(n-1, tmp, dst, src);
    }
}
```

Input:(n,A,B)=(3,1,3)

Output:

1→3

1→2

3→2

1→3

2→1

2→3

1→3

(2) 費布納契數列 (Fibonacci)：

I. 說明：

費氏數列中，第 n 項的值等於前兩項的和，而前兩項為 1。

設 $f(n)$ 表示第 n 項，根據定義： $f(n) = f(n-1) + f(n-2)$

請利用遞迴函數，求出 $f(n)$

II. 程式碼：

```
int f(int n)
{
    if (n==1 || n==2)
        return 1;
    else
        return f(n-1)+f(n-2);
}
```

六、儲存體的類別 (Storage Classes)

占用記憶體期間	類別	說明
動態	自動變數(auto)	<ul style="list-style-type: none">● auto 變數是在程式控制進入「所宣告的區塊」時才被產生出來，一離開此區塊，就會從記憶體中消失了。● 「區域變數」內定為自動變數。 例：auto float x;
	暫存器變數(register)	<ul style="list-style-type: none">● 宣告成 register 的變數會放到硬體暫存器中，減少運算時從記憶體載入的負擔。● register 變數只能用在「自動變數」上 例：register int counter = 1;
靜態	靜態變數(static)	<ul style="list-style-type: none">● static 變數從程式開始執行時便配置好儲存的記憶體，並設定初始值 (數值預設為 0，指標預設為 NULL)● static 變數包含內部靜態變數與外部靜態變數 例：static int count = 1;
	外部變數(extern)	<ul style="list-style-type: none">◆ 外部變數包含「全域變數」和「函式名稱」◆ 外部變數也屬於靜態變數，在程式開始執行時便佔有記憶體空間

七、範圍的規則 (Scope)

分類	說明	囊括
檔案範圍	可在正個檔案中使用	1. 全域變數(global variable) 2. 函式定義 3. 函式原形
區塊範圍	可在大括號{}包起來的範圍使用	1. 區域變數 (local Variable) 2. 巢狀區塊中的變數 3. 函式的參數
函式範圍	只能在函式的範圍中使用	標名： 1. goto 陳述句的標名 2. switch 中的 case
函式原型範圍		函式原形參數列中的識別字