

第 12 章 資料結構與程式設計

進入程式設計的領域

若禮看到表哥正在整理畢業旅行的衣物，就想幫表哥把一堆堆的衣服收進行李箱。表哥立刻阻止了他，並說：『這樣收拾衣服的方式不恰當喔！』

若禮覺得很奇怪，便問：『到底哪裡不恰當呢？』

表哥說：『我要將旅行第一天到最後一天要穿的衣服，由上到下依序疊放，這樣我就不用每天在行李箱裡翻找衣服了。這是我學資料結構得到的心得喔！』

表哥接著說：『資料結構主要是應用在電腦的資料處理上。但是，如果能將這些概念融會貫通，應用在日常生活中也很有意思。』

你知道什麼是資料結構嗎？你知道程式有哪些基本結構嗎？本章即將介紹幾種常見的資料結構，並說明程式的基本結構與結構化程式設計的方法。

- 12-1 認識資料結構 283
- 12-2 程式設計的基礎 293
- 12-3 結構化程式設計 322



軟體程式的開發與應用

12-1 認識資料結構

一個好的程式除了必須使用良好的演算法之外，也需使用適當的資料結構來組織資料，才能節省資料的儲存空間，並提昇資料處理的速度。本節將介紹資料結構的意義，以及幾種常見的資料結構。

12-1.1 資料結構簡介

資料結構（data structure）是用來組織及管理資料的結構設計。資料結構主要是在定義資料的放置方法及存取的規則。

常見的資料結構有**陣列**（array）、**堆疊**（stack）、**佇列**（queue）、**樹狀**（tree）、**鏈結串列**（linked list）…等結構；下面我們將分別介紹陣列、堆疊、及佇列等 3 種最基本的資料結構。

一點就通

資料結構可以用各式容器來比喻，容器有不同的形狀、大小、及存取方法；使用適當的容器可方便且有效地存取物品，如圖 12-1 所示。



▲ 圖 12-1 資料結構就像常見的各式容器

馬上練習

一、選擇題

- () 1. 下列何者的主要功能是在定義資料的放置方法及存取的規則？ (A)演算法 (B) 流程圖表示法 (C)敘述表示法 (D)資料結構。
- Ⓔ () 2. 下列何者不是一種資料結構（data structure）？ (A)佇列（queue） (B)堆疊（stack） (C)資料庫（database） (D)鏈結串列（linked list）。

12-1.2 陣列

一點就通

百貨公司的寄物櫃類似於陣列：
 "8 樓櫃" ⇒ 陣列名稱
 "081~088" ⇒ 陣列註標

陣列是由一群具有順序關係之相同類型資料元素所組成的資料結構。陣列中的資料元素通常都具有相同的「資料型別」，且多會依先後順序儲存在電腦的記憶體中。當我們在存取陣列中的資料元素時，要利用陣列的註標（index）來標示所要存取的資料元素。圖 12-2 為陣列的示意圖。



▲ 圖 12-2 一座寄物櫃相當於一個陣列

陣列的圖解說明

！小提示

多數的程式語言，如 Visual Basic，在宣告陣列時，陣列元素之預設值為 0。

陣列若依其註標之個數，可分為一維陣列、二維陣列、...、 n 維陣列；其中 n 表示註標之個數。圖 12-3 之 A (5) 是一個可儲存 5 個資料元素的一維陣列，B (3, 4) 是一個 3 列 4 行 ($3 \times 4 = 12$ 個資料元素) 的二維陣列示意圖。

一維陣列 A

陣列元素	0	0	0	0	0
陣列註標	(1)	(2)	(3)	(4)	(5)

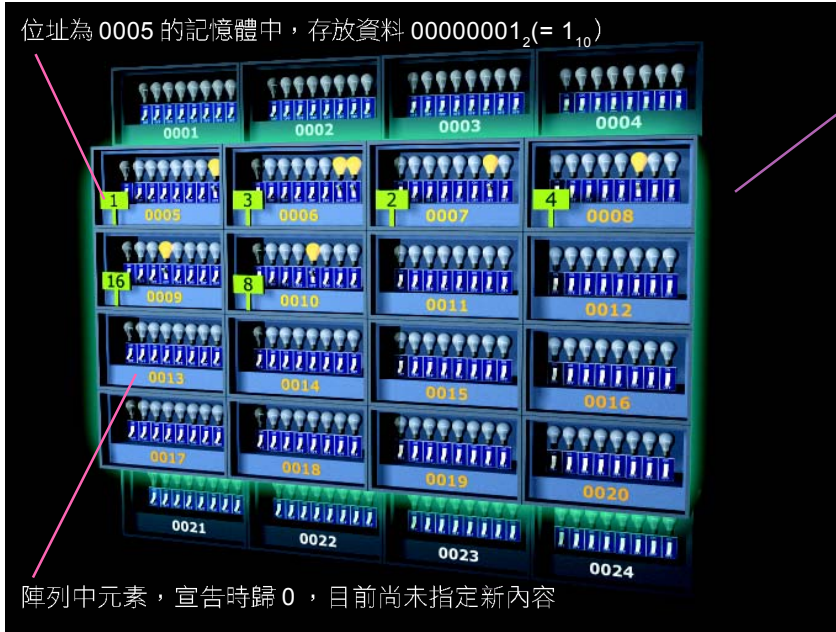
二維陣列 B

	第 1 行	第 2 行	第 3 行	第 4 行
第 1 列	0	0	0	0
陣列註標	(1, 1)	(1, 2)	(1, 3)	(1, 4)
第 2 列	0	0	0	0
陣列註標	(2, 1)	(2, 2)	(2, 3)	(2, 4)
第 3 列	0	0	0	0
陣列註標	(3, 1)	(3, 2)	(3, 3)	(3, 4)

二維陣列的註標 (m, n) 中
 m 為列數， n 為行數

▲ 圖 12-3 一維陣列及二維陣列

其實無論使用哪種資料結構，資料在記憶體中存放的基本方式都是一樣的：(1)以 0 與 1 記錄資料，(2)每單位之記憶體都有位址編號，如圖 12-4 所示。



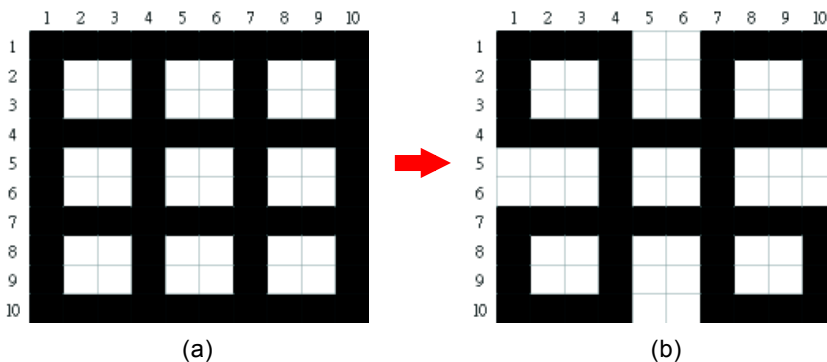
1. 宣告一維陣列：A(1 to 16)，指定元素內容為
 $A(1) = 1, A(2) = 3,$
 $A(3) = 2, A(4) = 4,$
 $A(5) = 16, A(6) = 8$
2. 宣告二維陣列：A(4, 4)，指定元素內容為
 $A(1, 1) = 1, A(1, 2) = 3$
 $A(1, 3) = 2, A(1, 4) = 4$
 $A(2, 1) = 16, A(2, 2) = 8$
 以上兩種陣列的宣告與應用，資料都將會以線性的方式存放在記憶體中（如圖中的 0005~0020）

▲ 圖 12-4 陣列存放資料於記憶體中的示意圖

陣列的練習－圖案變化遊戲

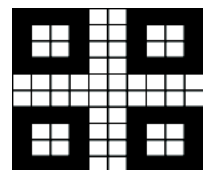
以下使用一個 10×10 的二維陣列來模擬資料的存取，改變 8 個元素（方塊）的值（顏色），可以將圖(a)中的 9 個小方形，變成 5 個小方形。

提示：將元素(1, 5)、(1, 6)、(5, 1)、(6, 1)、(5, 10)、(6, 10)、(10, 5)、(10, 6) 的顏色變為白色。



腦力激盪

如何改變圖(a)中的 16 個元素的值，使它的外觀由 9 小方形個變成 4 個小方形？（提示：答案不只一個，下圖是其中之一）



陣列的應用

陣列結構在程式設計上的應用相當廣泛，最為常見的應用是當程式中需使用多個相同資料型別的資料項目時，便可使用陣列結構來儲存及處理這些資料，使程式宣告敘述較為簡潔，如圖 12-5 所示。

說明：程式中需要處理 50 位同學的計概成績。

宣告一個可存放 50 個計概成績的陣列變數，只用一行程式敘述

```
Dim computer_grade(0 To 49) As Integer
```

宣告 50 個計概成績變數，共需 50 行程式敘述

```
Dim computer_grade1 As Integer
Dim computer_grade2 As Integer
Dim computer_grade3 As Integer
    .
    .
Dim computer_grade50 As Integer
```

▲ 圖 12-5 陣列（左）與多個變數宣告（右）的比較

！小提示

Visual Basic 之陣列註標值預設由 0 開始，如果不習慣陣列註標值與實際元素的個數相差 1 個；我們可以在一般宣告區加上一行 **Option Base 1** 敘述（如下圖所示），將註標之下限預設值設為 1，使陣列註標值等於陣列元素的個數。

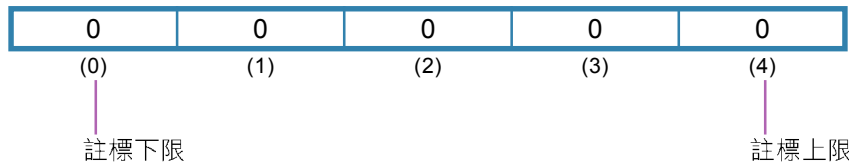


當然，圖 12-5（左）之後續程式的撰寫，也會因為利用到陣列結構而更容易。在 Visual Basic 中宣告一維陣列的語法為：

```
Dim 變數名稱（註標下限 To 註標上限） As 資料型別
```

例：Dim math_grade(0 To 4) As Integer

整數陣列名稱：math_grade



馬上練習

一、選擇題

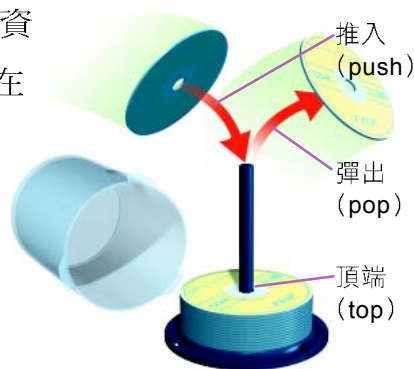
- Q () 1. 陣列 (array) 的註標 (index) 通常是 (A)實數 (B)整數 (C)小數 (D)分數。
- () 2. 下列關於陣列的敘述，何者錯誤？(A)陣列是一群資料型別相同的資料，通常儲存於電腦記憶體的連續位置裡 (B)陣列註標值預設為 0 (C)未指定陣列元素時，其陣列元素的初始值皆為 0 (D)利用 option base 敘述，可將註標的下限預設為任一個數字。

12-1.3 堆疊

堆疊是一種具有後進先出（Last In First Out, LIFO）特性的資料結構。資料存取原則是：當資料元素加入時，只能加到堆疊的頂端（top）；取出資料元素時，也會將放在頂端的資料優先取出。因此，從堆疊中取出的資料，會是最後加入至堆疊中的資料，如圖 12-6 所示。在堆疊中加入資料的動作稱為**推入**（push），取出資料的動作則稱為**彈出**（pop）。

一點就通

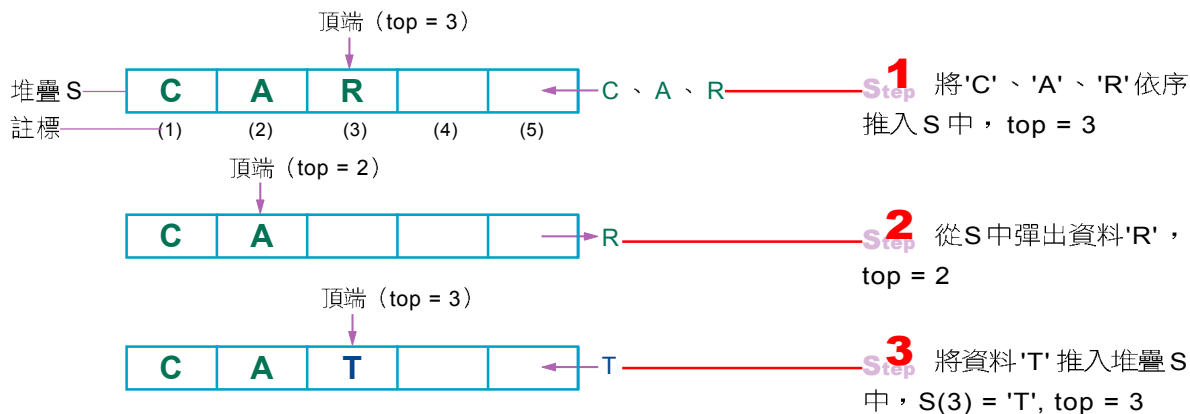
存放光碟的「布丁桶」類似堆疊，放入與取出都在最頂端發生，如圖 11-6 所示。



▲ 圖 12-6 堆疊資料結構後進先出示意圖

堆疊的圖解說明

基本上，堆疊可以用一個陣列，再加上一個頂端指標（top）來實作。以圖 12-7 為例，S 是一個可儲存五個資料元素的堆疊（陣列 S(1 to 5)），top 為頂端指標（記錄 S 的註標值）。

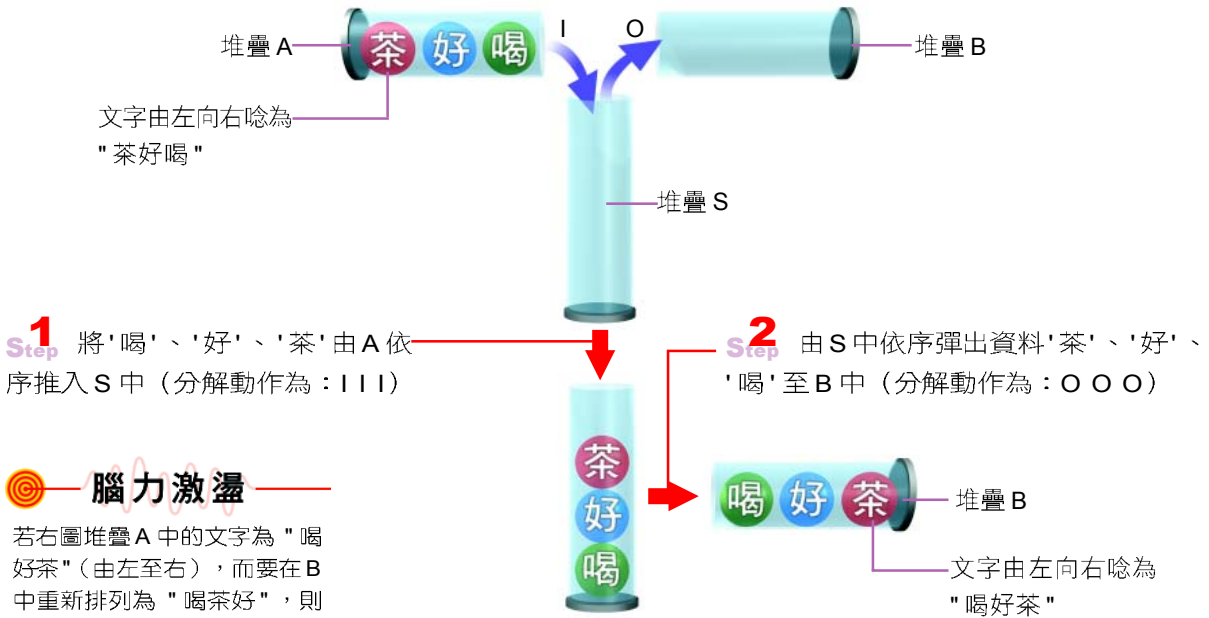


▲ 圖 12-7 堆疊的運作範例

堆疊的練習－文字重組遊戲

以下用一個可儲存 5 個資料元素的堆疊 S，及代表推入動作的程式 I、代表彈出動作的程式 O 來模擬堆疊的資料存取，如圖 12-8 所示：

- 說明：
1. S 只能接受從 A 推入的球，I 表示從 A 推入一球至 S
 2. S 只能彈出球至 B，O 表示從 S 彈出一球至 B
 3. 請利用 S 將 A 的文字 ("茶好喝") 於 B 重組成 "喝好茶"



▲ 圖 12-8 堆疊實作的圖解說明

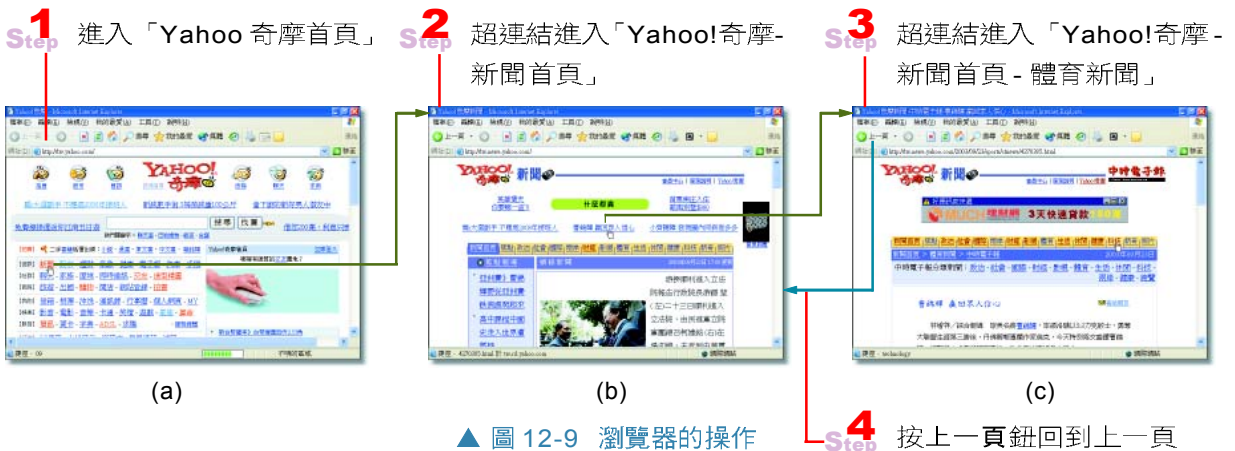
腦力激盪

若右圖堆疊 A 中的文字為 "喝好茶" (由左至右)，而要在 B 中重新排列為 "喝茶好"，則利用 I、O 程式的運算順序應為何？

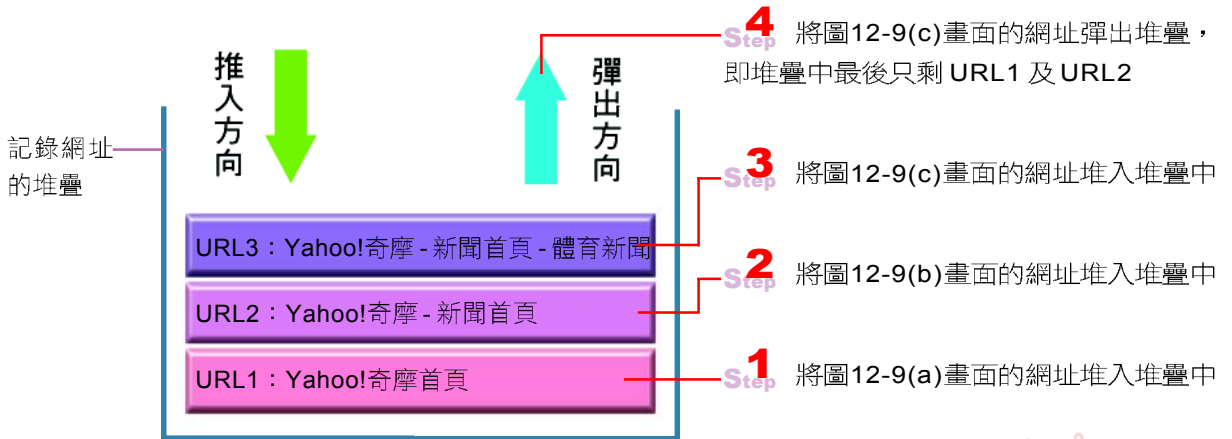
答：I O I I O O。

堆疊的應用

堆疊結構在程式設計上的應用相當廣泛，最常見的是瀏覽器的上一頁及下一頁功能，使用者按此二鈕可往返於曾經拜訪過的網址。圖 12-9 為使用 IE 瀏覽三個網頁，圖 12-10 為堆疊記錄網址的相關實例：



▲ 圖 12-9 瀏覽器的操作



▲ 圖 12-10 瀏覽器程式利用堆疊記錄網址的示意圖

腦力激盪

在圖 12-10 步驟 4 之後，若在瀏覽器上按下一頁鈕，會進入哪一個網頁？而網址堆疊中會剩下幾筆網址記錄呢？

馬上練習

一、選擇題

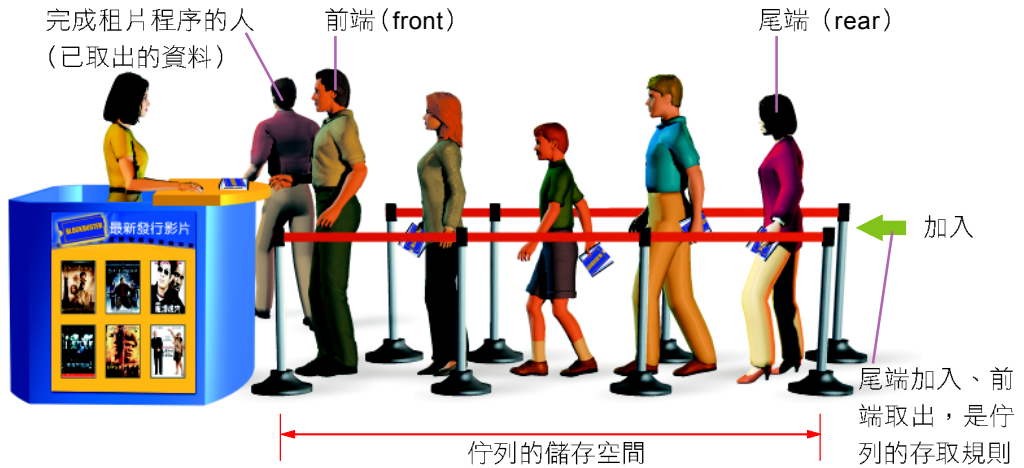
- Ⓔ () 1. 堆疊資料型態的基本特性是：(A)只進不出 (B)先進先出 (C)先進後出 (D)只出不進。
- Ⓔ () 2. 在堆疊中所有的新增和刪除的操作：(A)均在同一端完成 (B)均在不同的兩端完成 (C)必須在中間段完成 (D)在何處完成沒有限制。
- Ⓔ () 3. 一個堆疊原本是空的，再經過 push 7，push 6，pop，push 5，pop，push 4 後，堆疊最上層的內容為 (A)4 (B)5 (C)6 (D)7。

12-1.4 佇列

佇列是一種具有先進先出 (First In First Out, FIFO) 特性的資料結構。資料存取原則是：加入資料元素時，只能從佇列的尾端 (rear) 加入；取出資料元素時，則只能從佇列的前端 (front) 取出。因此，當要在佇列中取出資料時，最先加入的元素會被最先取出，圖 12-11 為佇列的範例。

一點就通

在十字路口等候綠燈要通行的車輛，也是一種佇列的應用。



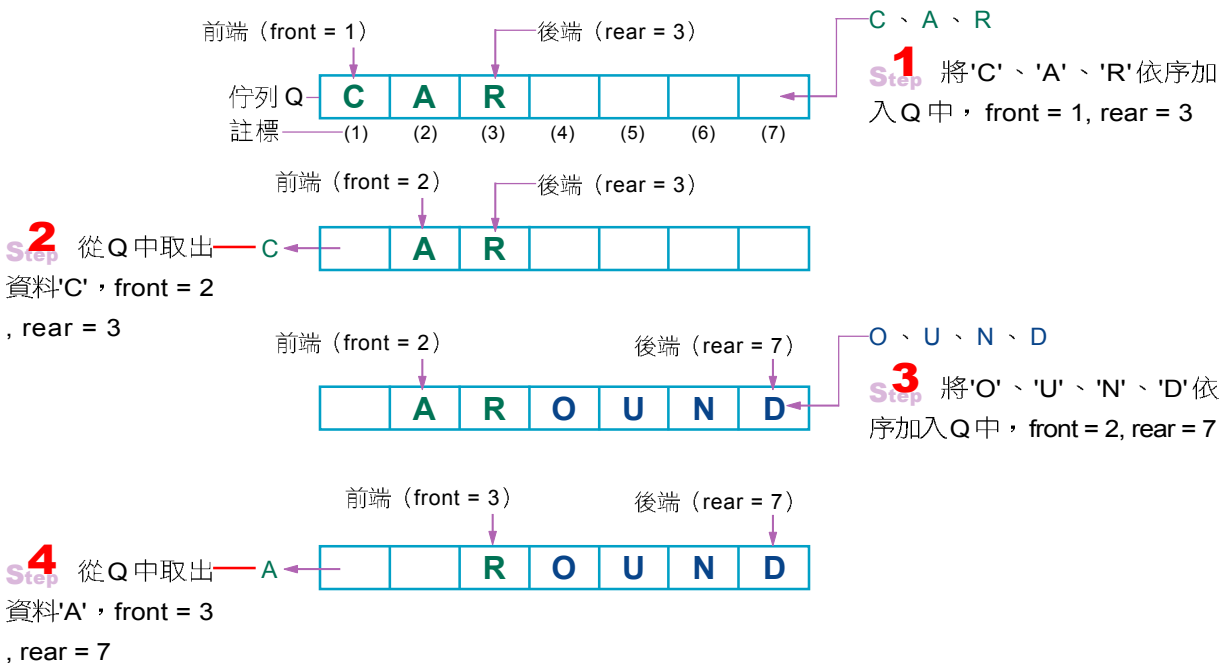
▲ 圖 12-11 佇列先進先出範例圖

佇列的圖解說明

！小提示

當圖 12-12 之佇列中的資料被取出後，即會空出原資料的儲存空間，如果程式中對於已取出資料的空間沒有做妥善的處理或再利用的話，就會有浪費儲存空間的缺點。

基本上，佇列可以用一個陣列，再加上二個指標（front, rear）來實作。以圖 12-12 為例，Q 是一個可儲存 7 個資料元素的佇列（陣列 Q(1 to 7)），front 及 rear 分別為前、後端指標（記錄 Q 的註標值）。

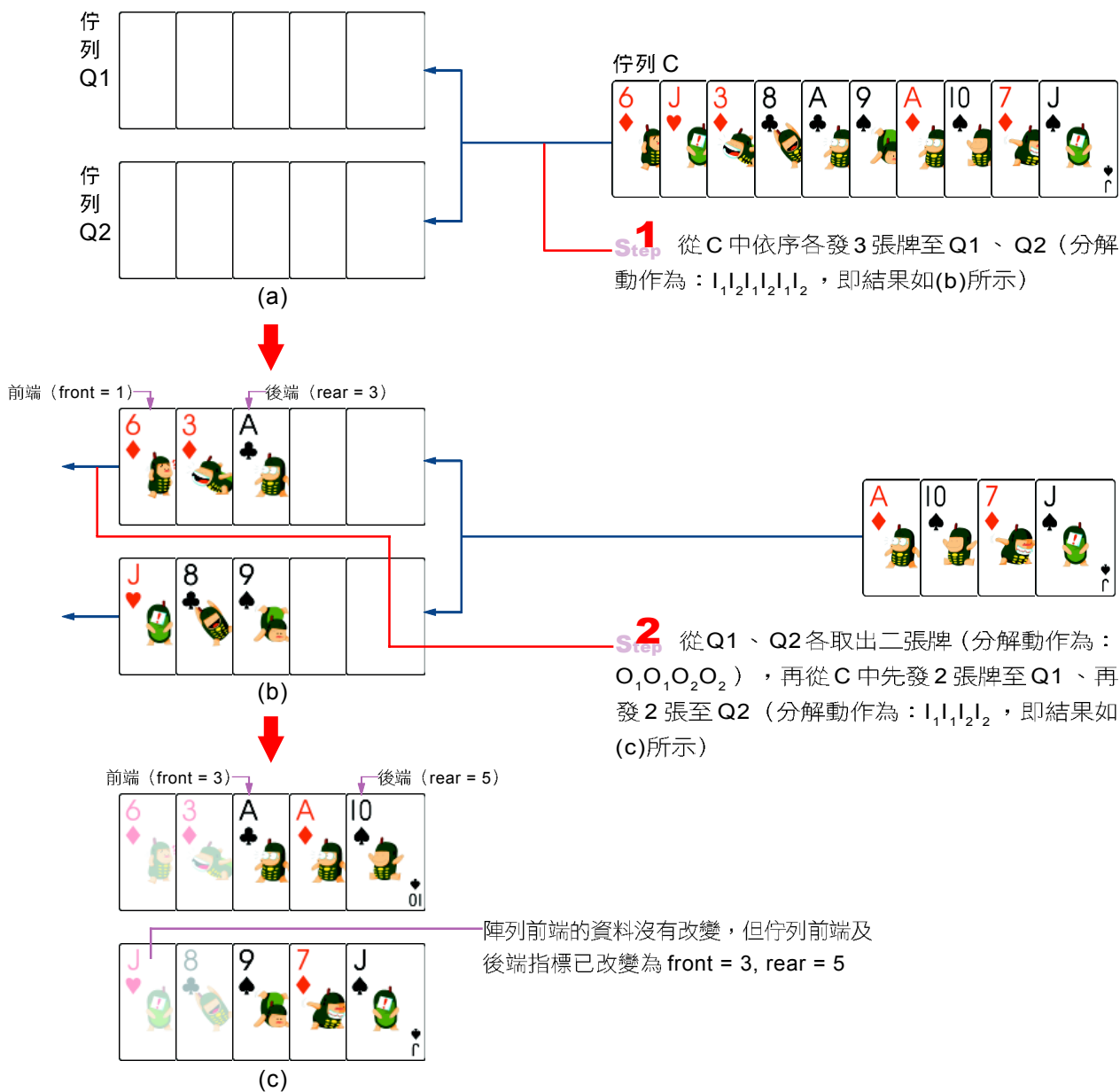


▲ 圖 12-12 佇列運作的圖解說明

佇列的練習－撲克牌遊戲

以下用三個各可儲存 5、5、10 個資料元素的佇列 (Q1, Q2, C)，及各代表加入動作的程式 I_1 、 I_2 ；代表取出動作的程式 O_1 、 O_2 ，來模擬佇列的資料存取，如圖 12-13 所示。

- 說明：
1. Q1 只能接受從 C 加入的牌， I_1 表示從 C 取出一張牌至 Q1。
 2. Q1 取出的牌捨棄不用， O_1 表示捨棄 Q1 的第一張牌。
 3. Q2、 I_2 、 O_2 的規則同上。



▲ 圖 12-13 佇列實作的圖解說明

佇列的應用

一點就通

銀行門口的號碼列印機，具有佇列應用的程式控制功能，符合先進先出的原則。

佇列結構在程式設計上的應用相當廣泛，最常見的是作業系統（如 Windows）中的印表機排程程式，如圖 12-14 所示。

使用者送出 5 份文件至印表機，此程式會依次序排列



文件名稱	狀態	擁有者	頁數	大小	已送交
Microsoft Word - 文件3	列印	sea	不適用	13.3 KB	下午 03:42:22 2003/6/21
Microsoft Word - 文件2	多工讓...	0047	不適用	13.3 KB	下午 03:42:42 2003/6/21
http://203.69.163.75/601.htm...	列印	USER	不適用	2.64 MB	下午 03:43:18 2003/6/21
Microsoft Word - 134a5博...	不適用	USER	不適用	65.3 KB	下午 03:43:44 2003/6/21
Microsoft Word - 校稿工...	不適用	DO14	不適用	84.9 KB	下午 03:44:16 2003/6/21

經過一段時間後，等待列印的文件只剩下 3 份，即最先進到佇列的 2 份文件已列印完畢



文件名稱	狀態	擁有者	頁數	大小	已送交
http://203.69.163.75/601.htm...	列印	USER	不適用	2.64 MB	下午 03:43:18 2003/6/21
Microsoft Word - 134a5博...	不適用	USER	不適用	65.3 KB	下午 03:43:44 2003/6/21
Microsoft Word - 校稿工...	不適用	DO14	不適用	84.9 KB	下午 03:44:16 2003/6/21

▲ 圖 12-14 佇列在印表機程式中的應用

馬上練習

一、選擇題

- Ⓔ () 1. 佇列資料型態的基本特性是：(A)只進不出 (B)先進先出 (C)先進後出 (D)只出不進。
- Ⓔ () 2. 在佇列中所有的新增和刪除的操作：(A)均在同一端完成 (B)在不同的兩端完成 (C)必須在中間段完成 (D)在何處完成沒有限制。

12-1 節練習

一、選擇題

- Ⓔ () 1. 在資料結構中，"FIFO" 稱為：(A)先進後出 (B)先進先出 (C)只進不出 (D)只出不進。
- Ⓔ () 2. 在資料結構中，何種存取方式是採用後進先出呢？(A)佇列 (B)堆疊 (C)環狀 (D)陣列。
- Ⓔ () 3. 下列哪一項非堆疊的事例？(A)排隊買票 (B)堆積木 (C)蓋房子 (D)堆盤子。
- Ⓔ () 4. 下列關於堆疊的敘述，何者正確？(A)存取方式為先進先出 (FIFO) (B)執行 MOV 指令時，會使用到堆疊 (C)執行 CALL 指令時，會使用到堆疊 (D)執行 JUMP 指令時，會使用到堆疊。
- Ⓔ () 5. 下列何者不正確？(A)陣列是給定一個固定的空間存放資料 (B)堆疊的處理方式是後進先出 (C)佇列的處理方式是後進先出 (D)佇列的處理方式是先進先出。

12-2 程式設計的基礎

利用程式語言可以開發出解決特定問題的程式，以便藉由電腦強大的運算處理能力，快速的完成原本費時、費力或不適合人工處理的工作。本節將介紹程式設計的基本概念，及組成程式的三種基本結構。

12-2.1 程式設計簡介

程式設計是軟體系統開發過程中的重要工作項目之一，其主要的工作內容是將使用者對軟體系統的需求撰寫為程式碼。下面將介紹程式設計的相關概念，以便同學們對程式設計的工作有一基本的認識。

程式設計的步驟

程式設計之目的是要透過程式語言，讓電腦協助我們解決一些特定的問題，其設計步驟與我們日常生活中的問題解決（problem solving）步驟相似，主要包括以下五項：

步驟 1：定義問題（Defining the Problem）

程式設計的第一步，必須先釐清問題，並預定問題解決後的結果。

步驟 2：擬定解題計畫（Planning the Solution）

問題定義後，接著須構思及草擬解決問題的方法，即解題的演算法，並以文字或圖示符號將解決的方法，逐步地記錄或描繪下來。

—— 小百科

程式設計師（programmer）是負責開發設計程式的人員；其工作內容除了撰寫程式碼之外，還須與其它參與程式設計的人員及使用者進行溝通協調…等工作。

—— 小提示

問題的解決方法可能不只一種，我們可從中選擇一種最佳的解決方法。

步驟 3：撰寫程式（Coding the Program）

使用程式語言，將前一步驟所擬定的解題計畫撰寫為程式碼。

小百科

程式錯誤的英文為 **bug**，若依字面上的意思可翻譯成「臭蟲」；其實這是有特別的典故！在電腦發展的早期，某個電腦工程師花了許多時間在尋找導致電腦系統當機的原因，最後竟然發現是因為一隻臭蟲掉進了電腦裡頭，才使得電腦當機。於是後來就有人將「電腦程式中的錯誤」及「去除程式錯誤」分別稱之為 **bug** 及 **debug**。

步驟 4：測試與除錯（Testing the Program）

將撰寫完成的程式，反覆地加以測試，並修正錯誤，直到程式可順利執行，且可與第一個步驟所預期的結果一致為止。

步驟 5：製作文件（Documenting the Program）

在上述四個程式設計的步驟中，程式設計者皆須製作相關的文件（例如：問題分析報告、程式流程圖、原始程式列表以及測試報告等文件），以便將來在維護程式時能有所依據。在程式開發完成後，也應對程式的使用者，提供詳細的操作手冊及相關說明文件，以便使用者能正確的使用程式。

以下是日常生活中「預約門診時間」，與設計簡易程式「找出數學成績 80 分以上的同學」之問題解決步驟對照表。

表 12-1 預約門診時間與找出數學成績 80 分以上同學的問題解決步驟對照表

問題 步驟	日常生活實例 預約門診時間（假設門診醫院已有病歷資料）	程式設計 找出數學成績 80 分以上的同學
定義問題	如何預約門診時間，並確定完成了預約手續	如何在全班數學成績中找出 80 分以上者？並印出其姓名
擬定解題計畫	<ol style="list-style-type: none"> 1. 取得門診時間表，查閱所要掛號的門診時間 2. 使用電話或網路掛號 3. 輸入身份證字號（或病歷號碼）、門診代號、及看病時間 4. 記錄就診序號及時間 	<ol style="list-style-type: none"> 1. 逐筆輸入每位同學的數學成績 2. 判斷成績是否 80 分以上 3. 如果符合條件，就列出同學的姓名
撰寫程式 （解決問題）	依照解題計畫，逐步進行預約門診時間的步驟	使用程式語言，將解題計畫中之問題解決方法撰寫為程式碼
測試與除錯 （測試結果）	利用門診預約查詢服務，檢查是否已完成預約的程序	彙集數學成績 80 分以上的考卷，以檢查電腦印出的同學姓名是否正確
製作文件	記錄就診序號及時間	將前述四個步驟中所撰寫的相關文件彙總整理，並留存備用

程式撰寫的原則

爲了設計出正確且易於閱讀與維護的程式，在程式設計時應儘量遵守下列幾點原則：

1. 養成宣告變數及常數的習慣：程式中所使用到的變數及常數都應事先宣告，以利於程式的撰寫、測試、與維護。
2. 遵照一定的命名規則：宣告變數及常數時，應該使用有意義的字元來命名，以利程式的閱讀及維護。例如：宣告一個存放國文成績的變數時，可將變數命名爲 "chinese_grade"。
3. 善用副程式或函數：對於需重複使用的程式敘述，應儘量寫成副程式或函數（將在 12-3.2 節中介紹），以避免重複撰寫，而使得程式過於冗長。
4. 善用內縮格式：程式語言中的某些敘述（例如 If 判斷敘述），須在開頭及結尾處使用特定的關鍵字將其中的程式敘述包圍起來；適當地將程式敘述內縮，可以提高程式的可讀性（如圖 12-15 所示）。
5. 儘量使用註解：在宣告變數及撰寫特定功能的程式敘述區塊旁，加上適當的註解說明，有助於程式的閱讀與維護。

小提示

在 Visual Basic 中，選按『工具/選項』選項，在開啟的選項/編輯器交談窗中，勾選要求變數宣告核取方塊，即可限定程式中所使用的變數皆需宣告。當程式中使用了未宣告的變數時，將出現如下的錯誤訊息。



內縮程式敘述

```
If a > b Then
    Print "a = ", a;
    Print "b = ", b
    Print "a 大於 b"
End If
```

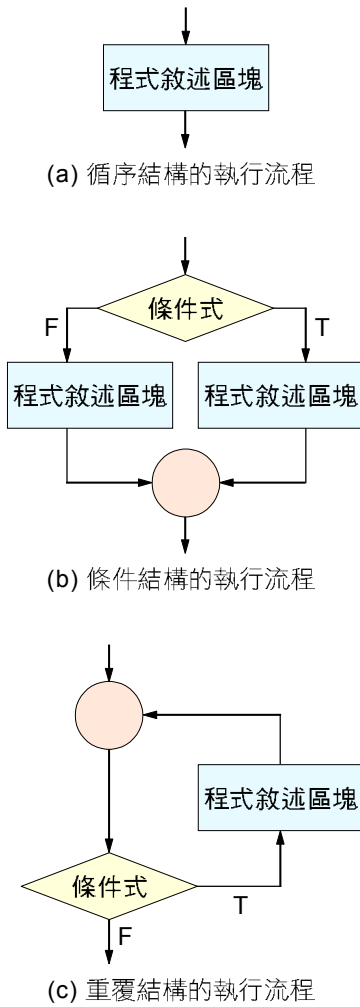
▲ 圖 12-15 善用內縮格式

小提示

程式碼很多時，在每一行程式敘述前加註行號，或使用空白行來隔開程式段落，也可增加程式的可讀性。爲了方便程式的解說，本書將在操作實例中加註行號，以利教師教學及同學閱讀。

程式的基本結構

一個程式大多是由**循序**（sequence）、**條件**（condition）、及**重覆**（repetition）等3種基本結構所組成。下面將分別介紹這3種基本結構。



▲ 圖 12-16 程式的基本結構

● **循序結構**（sequence structure）：由第一行的敘述開始，依序執行到程式的最後一行結束，是一種由上而下依序執行程式敘述的程式結構。其執行流程如圖 12-16(a)中的循序結構所示。

● **條件結構**（condition structure）：依照特定的條件或測試的結果來決定不同的執行流程，是一種利用分歧敘述來決定執行路徑的程式結構。其執行流程如圖 12-16(b)中的條件結構所示。

● **重覆結構**（repetition structure）：反覆執行迴路區塊，直到特定的條件出現為止，是一種利用迴圈重覆執行程式敘述的程式結構。其執行流程如圖 12-16(c)中的重覆結構所示。

循序、條件、及重覆等三種程式基本結構是由義大利科學家波因（Bohm）及傑克賓尼（Jacopini）所共同提出。

馬上練習

一、選擇題

- () 1. 下列何者不是程式設計過程中的工作項目之一？ (A)定義問題 (B)銷售原始碼 (C)撰寫程式 (D)製作文件。
- () 2. 下列何者不是程式設計時應遵守的原則？ (A)盡量使用註解 (B)善用內縮格式 (C)隨意命名變數名稱 (D)善用副程式及函數。
- @ () 3. 下列何者不是程式的基本結構？ (A)平行結構 (B)循序結構 (C)選擇結構 (D)重覆結構。

12-2.2 循序結構

循序結構是程式中最基本的結構，其執行順序是由程式的第一行敘述開始，然後逐步執行到最後一行。下面我們將按照 12-2.1 節所介紹的程式設計步驟，說明如何利用循序結構來撰寫一個可以讓使用者更換並預覽不同手機機殼的程式。

程式設計實例－循序結構

步驟 1：定義問題

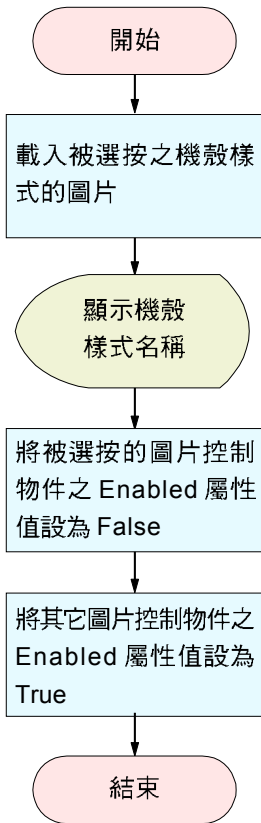
展示套用不同機殼的手機外貌，讓使用者在按下某一機殼圖片時，可以顯示該機殼樣式的手機圖片及名稱（如圖 12-17 所示）。



▲ 圖 12-17 「手機機殼變裝秀」程式之執行結果

步驟 2：擬定解題計畫

- (1) 利用數個圖片控制物件讓使用者可以選按不同機殼樣式的圖片。
- (2) 依照使用者選按之機殼樣式，調整標籤控制物件之 Caption 屬性，讓使用者可以查看其選按之機殼樣式的名稱。



- (3) 利用 LoadPicture() 函數載入使用者選按之機殼樣式的圖片。
- (4) 調整圖片控制物件的 Enabled 屬性，讓使用者不能重複選按同一個圖片控制物件。
- (5) 本實例之解題步驟，若以流程圖來表示，可以繪製如圖 12-18 所示。本例當使用者按一下某一機殼圖片時，程式會如流程圖所示，由上到下依序執行；此即循序結構。

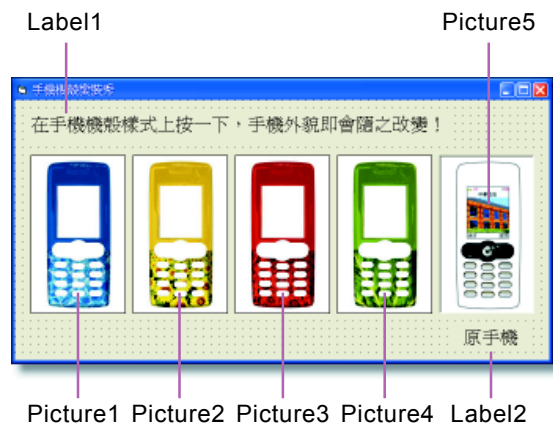
步驟 3：撰寫程式

在撰寫程式之前，我們須先建立如圖 12-19 所示的表單外觀，並依照表 12-2 設定各控制物件的屬性。

▲ 圖 12-18 手機機殼變裝秀之流程圖

表 12-2 手機機殼變裝秀之控制物件屬性設定

控制物件	屬性	設定值
Form1	Caption	" 手機機殼變裝秀 "
Label1	Caption	" 在手機機殼樣式上按一下，手機外貌即會隨之改變！ "
	Font	字型大小設為 "14"
Label2	Alignment	"2" (置中對齊)
	Caption	" 原手機 "
	Font	字型大小設為 "14"
Picture1	Appearance	"0" (平面)
	Picture	" 手機機殼 1.jpg "
Picture2	Appearance	"0" (平面)
	Picture	" 手機機殼 2.jpg "
Picture3	Appearance	"0" (平面)
	Picture	" 手機機殼 3.jpg "
Picture4	Appearance	"0" (平面)
	Picture	" 手機機殼 4.jpg "
Picture5	Picture	" 原手機.jpg "



▲ 圖 12-19 手機機殼變裝秀之表單外觀

操作實例 1 「手機機殼變裝秀」程式設計－撰寫程式碼

設計一個可以讓使用者更換手機殼機的循序結構程式。

10~60 敘述區塊為循序結構

```

Private Sub Picture1_Click()
10 Picture5.Picture = LoadPicture(App.Path & \"手機1.jpg\")
20 Label2.Caption = \"水紋機殼\"
30 Picture1.Enabled = False
40 Picture2.Enabled = True
50 Picture3.Enabled = True
60 Picture4.Enabled = True
End Sub

```

加註行號是為了解說方便，在撰寫程式時，可視實際需要，決定是否加入

Step 1 在 Picture1 的 Click 事件程序中撰寫更換手機圖片的程式碼

1.1 輸入程式碼 10，設定 Picture5 要載入的圖檔

1.2 輸入程式碼 20，設定使用者選按之機殼樣式的名稱

1.3 輸入程式碼 30~60，設定使用者無法再選按 Picture1，但可選按 Picture2~4

```

Private Sub Picture2_Click()
10 Picture5.Picture = LoadPicture(App.Path & \"手機2.jpg\")
20 Label2.Caption = \"向日葵機殼\"
30 Picture1.Enabled = True
40 Picture2.Enabled = False
50 Picture3.Enabled = True
60 Picture4.Enabled = True
End Sub

```

循序結構

Step 2 在 Picture2~4 的 Click 事件程序中撰寫更換手機圖片的程式碼

2.1 選取 Picture1 的 Click 事件程序中的程式碼 10~60，按複製鈕複製該段程式碼

2.2 分別切換至 Picture2~4 的 Click 事件程序，並按貼上鈕，貼上剛才複製的程式碼

```

Private Sub Picture3_Click()
10 Picture5.Picture = LoadPicture(App.Path & \"手機3.jpg\")
20 Label2.Caption = \"玫瑰機殼\"
30 Picture1.Enabled = True
40 Picture2.Enabled = True
50 Picture3.Enabled = False
60 Picture4.Enabled = True
End Sub

```

循序結構

2.3 修改程式碼，設定 Picture5 所要載入的圖檔、Label2 所要顯示的機殼樣式名稱、及 Picture1~4 選按是否有效

```

Private Sub Picture4_Click()
10 Picture5.Picture = LoadPicture(App.Path & \"手機4.jpg\")
20 Label2.Caption = \"葉子機殼\"
30 Picture1.Enabled = True
40 Picture2.Enabled = True
50 Picture3.Enabled = True
60 Picture4.Enabled = False
End Sub

```

循序結構


步驟 4：測試與除錯


在操作實例 2 中，我們將測試程式是否會在使用者按一下某一機殼圖片時，更換原手機圖片為使用者選按之機殼樣式的手機圖片，並在表單上顯示該機殼樣式的名稱。如果程式執行有誤，則須進行除錯工作，直到測試無誤為止。

操作實例 2

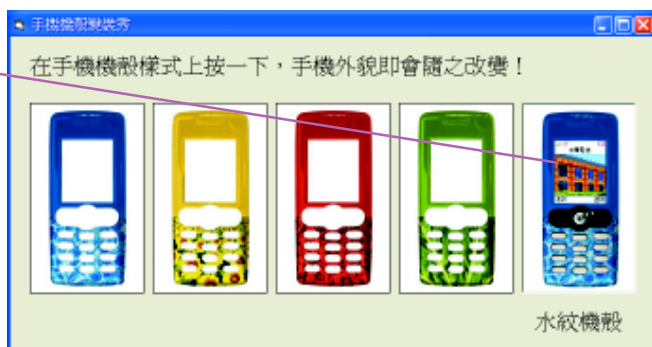
「手機機殼變裝秀」程式設計－執行程式

按一下某一機殼圖片，測試原手機之機殼樣式是否會更換為選按之機殼樣式的手機圖片，並顯示該機殼樣式的名稱。

Step 1 按開始鈕 ，執行程式

Step 2 按此手機機殼樣式 

手機圖片換成套用「水紋」機殼樣式的圖案了



！小提示

「測試及除錯」與「製作文件」是程式設計過程中，相當重要的步驟。但為簡潔考量，在 12-2.3 節、12-2.4 節及 12-3.2 節中之程式設計實例，將不再針對此二步驟作詳細的說明，同學可自行類推練習。

步驟 5：製作文件

文件製作是程式設計的工作項目之一，本步驟必須將前面四個步驟所製作的文件加以彙總整理並留存，以備日後程式維護使用，同時也必須製作可供使用者參照的操作手冊，或相關說明文件。

馬上練習

一、實作題

1. 請開啓已佈妥控制物件的範例檔案 "Ex06-a.vbp"，利用循序結構設計一個可計算男、女生標準體重的程式，最後將檔案儲存爲 "Sp6-a.vbp"。

提示：(1) 利用 `InputBox()` 函數讀取使用者輸入之身高。

(2) 男生的標準體重計算公式：
(身高 - 80) × 0.7。

(3) 女生的標準體重計算公式：
(身高 - 70) × 0.6。



12-2.3 條件結構

條件結構是一種利用條件判斷敘述來選擇程式執行路徑的程式結構。以下分別說明兩種常用的條件判斷敘述－If 判斷敘述與 Select Case 敘述。

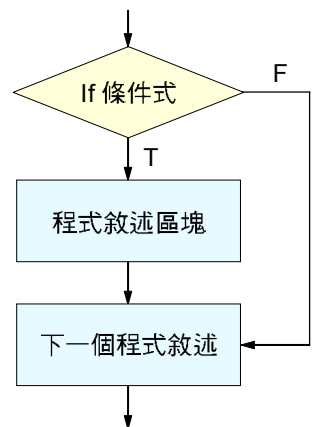
If 判斷敘述

If 判斷敘述可再細分爲 **If-Then 敘述** 及 **If-Then-Else 敘述** 兩種，分別說明如下：

- **If-Then 敘述**：用來判斷條件式成立時，所要執行的程式敘述，其執行流程若以流程圖表示，可繪製如圖 12-20 所示。一般常依條件式成立時所要執行的程式敘述多寡，分爲「單行式 If-Then 敘述」及「區塊式 If-Then 敘述」，其語法說明如下。

小提示！

第十二章之程式語法說明中，將以「程式敘述區塊」來代表一個（含）以上的程式敘述。



▲ 圖 12-20 If-Then 敘述的流程圖

語法 1： If 條件式 Then 程式敘述

語法 2： If 條件式 Then
程式敘述區塊
End If

說明：

1. 語法 1 為單行式 If-Then 敘述，當「條件式」成立時，將執行 Then 之後的程式敘述，否則就繼續執行下一個程式敘述。
2. 語法 2 為區塊式 If-Then 敘述，當「條件式」成立時，會執行 Then 之後的程式敘述區塊。
3. 區塊式 If-Then 敘述必須與 End If 成對使用。

```
score1 = 68
score2 = 75
If score2 >= score1 Then _
    Print "成績持平或進 _
        步了喔!"
If score2 < score1 Then _
    Print "成績退步了喔!"
```

圖 12-21 為使用 If-Then 敘述的程式片段實例，此範例會判斷第 2 次段考成績（score2）是否高於第 1 次段考成績（score1），並在螢幕上輸出成績進步或退步

▲ 圖 12-21 If-Then 敘述的實例的訊息。

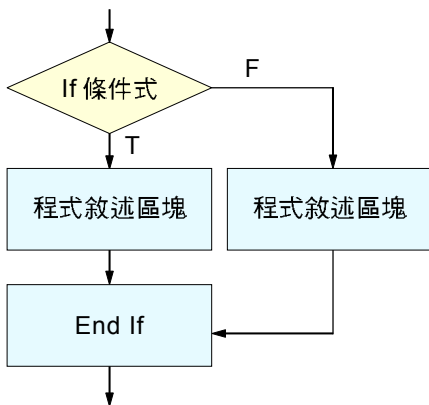
● **If-Then-Else 敘述：**用來判斷條件式成立時，所要執行的程式敘述區塊，以及條件式不成立時，須執行的程式敘述區塊，其執行流程若以流程圖表示，可繪製如圖 12-22 所示。If-Then-Else 敘述的語法說明如下：

語法：

If 條件式 Then
條件式為真（True）程式敘述區塊
Else
條件式為假（False）程式敘述區塊
End If

說明：

1. 當「條件式」成立時，會執行 Then 之後的程式敘述區塊；當「條件式」不成立時，會執行 Else 之後的程式敘述區塊。
2. If-Then-Else 敘述必須與 End If 成對使用。



▲ 圖 12-22 If-Then-Else 敘述的流程圖

前述使用 If-Then 敘述來判斷段考成績是否進步之實例（P.300 的圖 12-21），若改用 If-Then-Else 敘述來改寫，則改寫後的程式敘述會變得較有層次（如圖 12-23 所示）。

```
score1 = 68
score2 = 75
If score2 >= score1 Then
    Print " 成績持平或進
        步了喔!"
Else
    Print " 成績退步了喔!"
End If
```

▲ 圖 12-23 使用 If-Then-Else 敘述的實例

巢狀 If 判斷敘述

在程式設計實務中，當抉擇不只兩種時，我們可以使用**巢狀 If 判斷敘述**來處理。所謂**巢狀 If 判斷敘述**，是指 If 判斷敘述中還有 If 判斷敘述，例如圖 12-24 所示的程式，便是一個依據年齡來判斷可觀賞影片等級的巢狀 If 判斷敘述實例。

```
If age < 6 Then
    Print " 僅能觀賞普遍級的影片 "
Else
    If age < 12 Then
        Print " 只能觀賞普遍級及保護級的影片 "
    Else
        If age < 18 Then
            Print " 只能觀賞普遍級、保護級、及輔導級的影片 "
        Else
            Print " 各級影片皆能觀賞 "
        End If
    End If
End If
```

▲ 圖 12-24 使用巢狀 If 判斷敘述的實例

程式設計實例－巢狀 If 判斷敘述

步驟 1：定義問題

隨機產生一個介於 1~13 之間的亂數，讓使用者猜測亂數值是否大於 7，然後顯示猜對與否的訊息（如圖 12-25 所示）。



▲ 圖 12-25 「比大小遊戲」程式之執行結果

腦力激盪

如果要使用 If-Then-Else 敘述來判斷成績是否及格，則程式敘述該怎麼撰寫呢（假設 score 是用來儲存成績的變數）？

小百科

Visual Basic 也提供了一種稱為 If-ElseIf 的巢狀判斷敘述。其語法如下所示：

```
If 條件式 1 Then
    條件式 1 為真 (True) 之程式敘述
ElseIf 條件式 2 Then
    條件式 2 為真 (True) 之程式敘述
.
.
.
ElseIf 條件式 N Then
    條件式 N 為真 (True) 之程式敘述
End If
```

小提示

當程式敘述太長而無法撰寫於一行時，可使用底線(_)來接續程式。

步驟 2：擬定解題計畫

- (1) 表單載入時，利用 `Rnd()` 函數及 `Int()` 函數隨機產生一個介於 1~13 之間的亂數（正整數）；表 12-3 是 `Rnd()` 函數及 `Int()` 函數的用法說明。

表 12-3 `Rnd()` 函數及 `Int()` 函數的用法說明

函數	說明
<code>Rnd()</code> 函數	<code>Rnd()</code> 函數可用來產生一個 ≥ 0 且 < 1 的亂數 例如： <code>Rnd() * 3</code> 敘述會產生一個介於 0 與 3 之間的亂數
<code>Int()</code> 函數	<code>Int()</code> 函數可用來取小於或等於某一數值的最大整數值 例如： <code>Print Int(3.5)</code> 敘述的輸出結果為整數 3

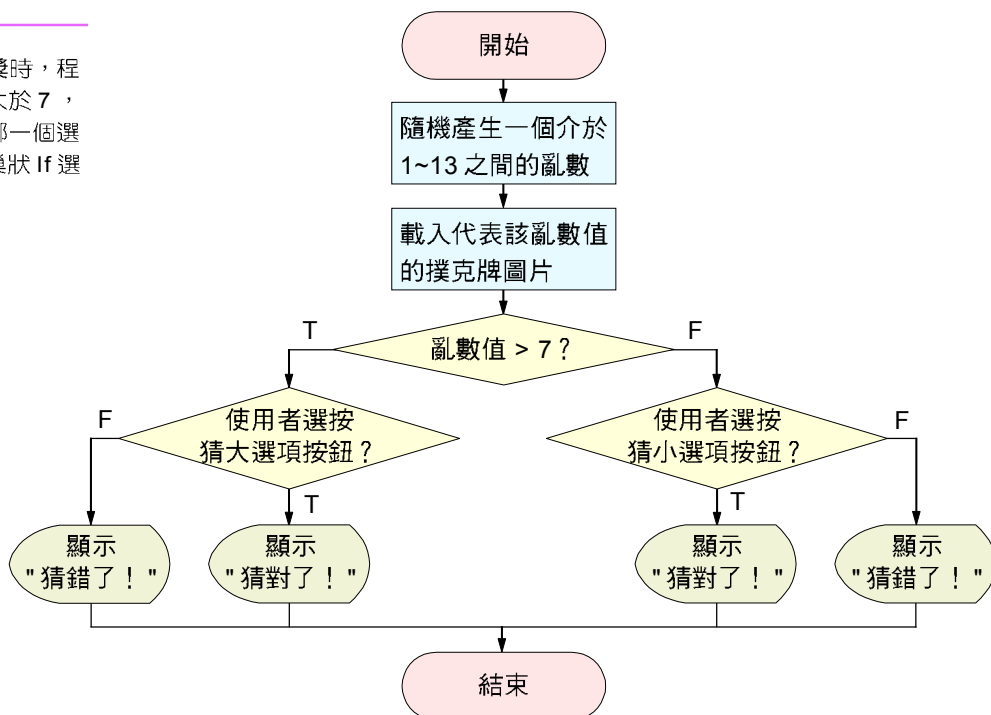
 腦力激盪

在即時運算視窗中執行 "Print `Int(Rnd() * 5) + 1`" 程式敘述，則螢幕輸出之結果為何？

- (2) 當使用者按開獎鈕時，載入代表亂數值的撲克牌圖片，然後利用巢狀 `If` 判斷敘述判斷亂數值是否大於 7，並顯示猜對與否的訊息。
- (3) 當使用者按再玩一次鈕時，重新產生一個介於 1~13 之間的亂數值，讓使用者可以繼續玩遊戲。
- (4) 本實例之解題步驟，若以流程圖來表示，可以繪製如圖 12-26 所示。

! 小提示

本例當使用者按下開獎時，程式會先判斷亂數是否大於 7，再判斷使用者選按了哪一個選項，是否猜對；此即巢狀 `If` 選擇結構。



▲ 圖 12-26 比大小遊戲之流程圖

步驟 3：撰寫程式

爲了節省同學們在表單上佈建控制物件的時間，請同學開啓已佈建控制物件的範例檔案 "Ex06-2b.vbp"（如圖 12-27 所示），並依照操作實例 3 的說明，撰寫「比大小遊戲」的程式。



▲ 圖 12-27 範例檔案 Ex06-2b.vbp

操作實例 3 「比大小遊戲」程式設計

撰寫一個「比大小遊戲」的程式，然後測試程式執行的結果。

由於 "Int(Rnd() * 13)" 所產生的整數介於 0~12 之間，因此必須加 1，使 n 值可介於 1~13 之間

```

Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
10 Dim n As Integer '變數n用來儲存亂數值
20 n = Int(Rnd() * 13) + 1
30 Picture1.Picture = LoadPicture(App.Path & "紅心" & n & ".jpg")
40 If n > 7 Then
50 If Option1.Value = True Then
60 MsgBox "猜對了!", "答案揭曉"
70 Else
80 MsgBox "猜錯了!", "答案揭曉"
90 End If
100 Else
110 If Option2.Value = True Then
120 MsgBox "猜對了!", "答案揭曉"
130 Else
140 MsgBox "猜錯了!", "答案揭曉"
150 End If
160 End If
End Sub
    
```

40~160 敘述區塊為選擇結構

```

Project1 - Form1 (程式碼)
Command3 Click
Private Sub Command2_Click()
10 Picture1.Picture = LoadPicture(App.Path & "未知數字.jpg")
End Sub
    
```

Step 1 在 Command1 的 Click 事件程序中撰寫產生亂數及判斷使用者猜中與否的程式碼

1.1 輸入程式碼 10~20，宣告變數 n，並設定其為隨機產生之 1~13 之間的整數

1.2 輸入程式碼 30，設定 Picture1 要載入的圖檔

1.3 輸入程式碼 40~90，設定變數 n 值大於 7 時，顯示猜對及猜錯的訊息

1.4 輸入程式碼 100~160，設定變數 n 值小於或等於 7 時顯示，猜對及猜錯的訊息

Step 2 在 Command2 的 Click 事件程序中撰寫載入問號圖檔的程式碼

2.1 輸入程式碼 10，設定 Picture1 要載入的圖檔

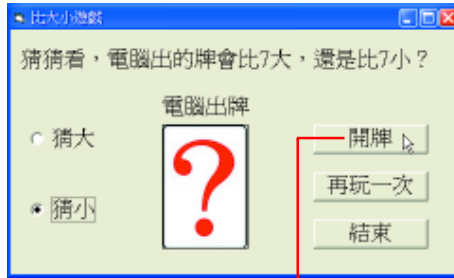
3 執行程式，測試程式執行的結果

3.1 按開始鈕 ▶ 執行程式

3.2 按猜小鈕再按開牌鈕

3.3 按確定鈕，關閉此交談窗

3.4 按結束鈕，結束程式的執行

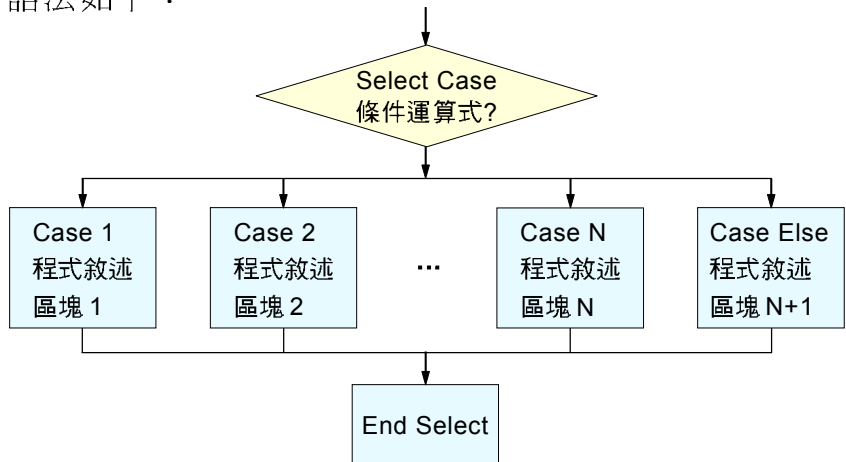


顯示了猜錯的訊息交談窗

顯示代表電腦隨機產生之亂數值的撲克牌圖案

Select Case 判斷敘述

Select Case 敘述是專門用來處理多重選擇的條件判斷敘述，其執行流程若以流程圖表示，可繪製如圖 12-28 所示。利用 Select Case 判斷敘述，程式可以設定多個執行路徑，並依設定的條件來控制程式的執行流向，其語法如下：



▲ 圖 12-28 Select Case 判斷敘述的流程圖

```

語法： Select Case 資料或運算式
        Case 條件式 1
            程式敘述區塊 1
        Case 條件式 2
            程式敘述區塊 2
            .
            .
            .
        Case 條件式 N
            程式敘述區塊 N
        Case Else
            程式敘述區塊 N+1
    End Select

```

說明：

1. 若資料或運算式符合「條件式 1」時，則執行「程式敘述區塊 1」；若符合「條件式 2」，則執行「程式敘述區塊 2」；…，依此類推；若資料或運算式符合「條件式 N」時，則執行「程式敘述區塊 N」。
2. 若不符合任何條件式，則執行「程式敘述區塊 N+1」。

前述使用巢狀 If 判斷敘述來判斷各年齡層可觀賞影片等級之實例（P.301 的圖 12-24），若改用 Select Case 判斷敘述來改寫，則改寫後的程式敘述會變得較為簡潔（如圖 12-29 所示）。

```

Select Case age
    Case age < 6
        Print "僅能觀賞普遍級的影片 "
    Case age < 12
        Print "只能觀賞普遍級及保護級的影片 "
    Case age < 18
        Print "只能觀賞普遍級、保護級、及輔導級的影片 "
    Case Else
        Print "各級影片皆能觀賞 "
End Select

```

▲ 圖 12-29 使用 Select Case 判斷敘述的實例

小提示 !

Select Case 判斷敘述中的 Case 敘述可有下列幾種語法表示：

1. 數值或字串：例如 Case 1 、 Case 2, 3, 4 、或 Case "A", "B" 。
2. 指定範圍：例如 Case 1 To 10 。
3. 關係運算式：例如 Case Is > 2 、 Case Is = 1 。

小提示 !

雖然巢狀 If 判斷敘述也可用來處理多抉擇的條件判斷，但程式往往會因層次過多而顯得冗長不易閱讀，且容易因疏忽而造成 If 起始敘述與 End If 結尾敘述不對稱的情形；因此，在遇到多抉擇的條件判斷時，應儘可能使用 Select Case 判斷敘述。

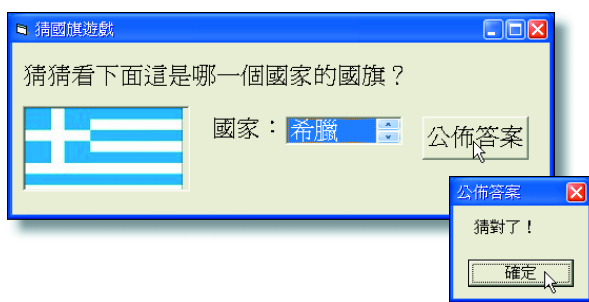
程式設計實例－ Select Case 判斷敘述

步驟 1：定義問題

腦力激盪

如果要使用 **Select Case** 判斷敘述來判斷成績等第（即 90 分以上為優等，80 分以上為甲等，70 分以上為乙等，60 分以上為丙等，60 分以下為劣等），則程式敘述該怎麼撰寫呢（假設 **score** 是用來儲存成績的變數）？

隨機載入一張國旗圖片，讓使用者猜測國家名稱，然後顯示猜對與否的訊息（如圖 12-30）。



▲ 圖 12-30 「猜國旗遊戲」程式之執行結果

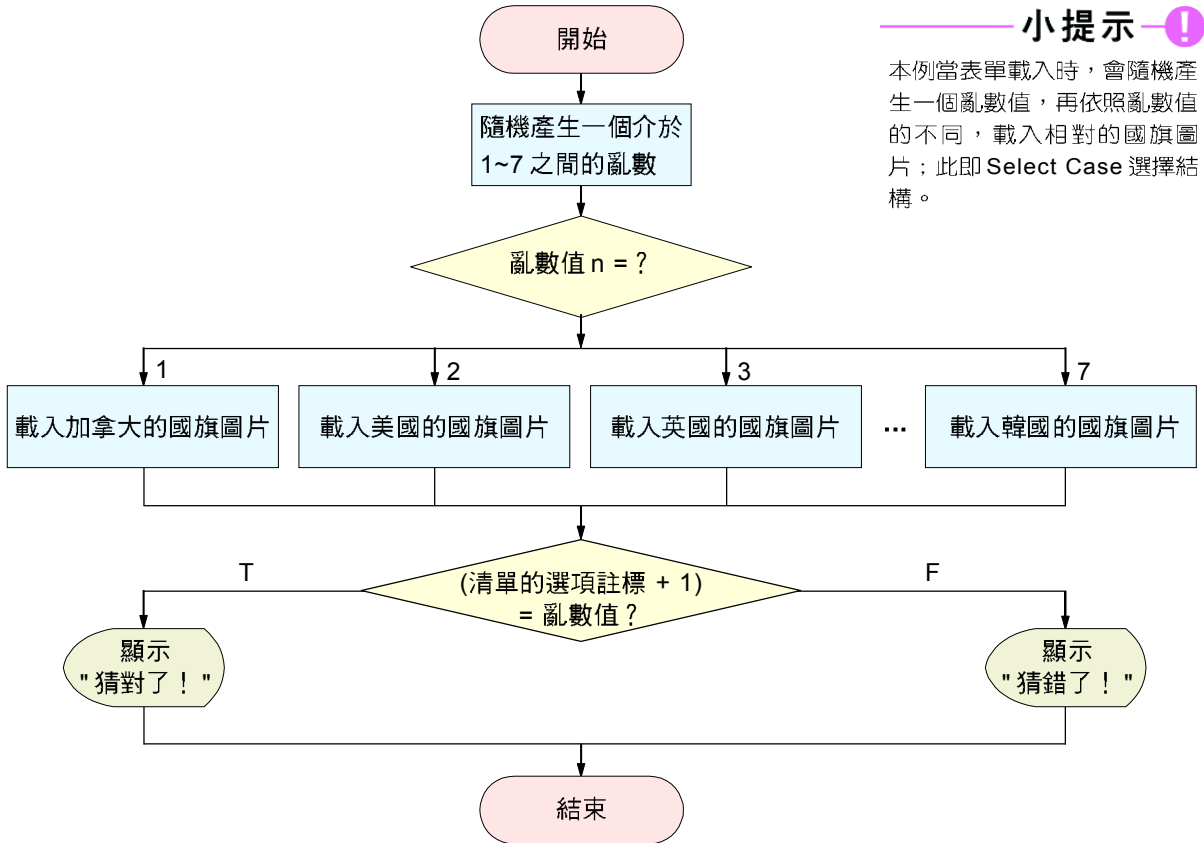
步驟 2：擬定解題計畫

- (1) 表單載入時，利用 **Rnd()** 函數及 **Int()** 函數隨機產生一個 1~7 之間的亂數（由於本實例共提供 7 面國旗圖片，因此將隨機產生 1~7 之間的亂數）。
- (2) 利用 **Select Case** 判斷敘述，依據亂數值載入相對應的國旗圖片。
- (3) 當使用者按**公佈答案**鈕時，利用 **If-Then-Else** 判斷敘述，判斷答案是否正確並顯示猜對與否的訊息。
- (4) 利用 **ListBox** 控制物件的 **ListIndex** 屬性來取得使用者選按的國名選項。表 12-4 是 **ListIndex** 屬性的用法說明。

表 12-4 ListBox 控制物件的 ListIndex 屬性用法說明

屬性	說明
ListIndex	用來表示清單中被選取項目的註標；第 1 個項目的註標值為 0

(5) 本實例之解題步驟，若以流程圖來表示，可以繪製如圖 12-31 所示。



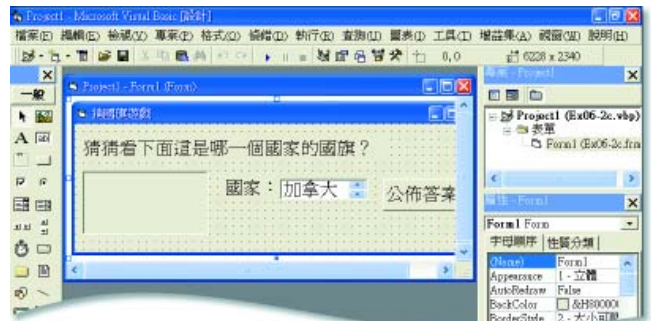
小提示!

本例當表單載入時，會隨機產生一個亂數值，再依照亂數值的不同，載入相對的國旗圖片；此即 Select Case 選擇結構。

▲ 圖 12-31 猜國旗遊戲之流程圖

步驟 3：撰寫程式

為了節省同學們在表單上佈建控制物件的時間，請同學開啓已佈建控制物件的範例檔案 "Ex06-2c.vbp"（如圖 12-32 所示），並依照操作實例 4 的說明，撰寫「猜國旗遊戲」的程式。



▲ 圖 12-32 範例檔案 Ex06-2c.vbp

操作實例 4

「猜國旗遊戲」程式設計

撰寫一個「猜國旗遊戲」的程式，然後測試程式執行的結果。

s.1 在一般程序中，宣告變數 n 為整數資料型別，並在 Form 的 Load 事件程序中撰寫隨機產生亂數及載入國旗圖片的程式碼

1.1 輸入程式碼 10，設定變數 n 為隨機產生之 1 至 7 的整數

1.2 輸入程式碼 20~30，設定變數 n 值不同時，Picture1 所要載入的圖檔

```
Project1 - Form1 (程式碼)
Form Load
Dim n As Integer '變數n用來儲存亂數值
Private Sub Form_Load()
10 n = Int(Rnd * 7) + 1
20 Select Case n
Case 1: Picture1.Picture = LoadPicture(App.Path & "\加拿大.jpg")
Case 2: Picture1.Picture = LoadPicture(App.Path & "\美國.jpg")
Case 3: Picture1.Picture = LoadPicture(App.Path & "\英國.jpg")
Case 4: Picture1.Picture = LoadPicture(App.Path & "\德國.jpg")
Case 5: Picture1.Picture = LoadPicture(App.Path & "\希臘.jpg")
Case 6: Picture1.Picture = LoadPicture(App.Path & "\義大利.jpg")
Case 7: Picture1.Picture = LoadPicture(App.Path & "\韓國.jpg")
30 End Select
End Sub
```

20~30 敘述區塊為選擇結構

s.2 在 Command1 的 Click 事件程序中撰寫判斷答案是否正確的程式碼

2.1 輸入程式碼 10~20，宣告變數 i 為整數資料型別，並設定變數 i 為 List1 清單控制物件中被選取項目的註標值加 1

2.2 輸入程式碼 30~70，設定猜對及猜錯時，所要顯示的訊息交談窗

```
Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
10 Dim i As Integer
20 i = List1.ListIndex + 1
30 If i = n Then
40 MsgBox "猜對了!", "公佈答案"
50 Else
60 MsgBox "猜錯了!", "公佈答案"
70 End If
End Sub
```

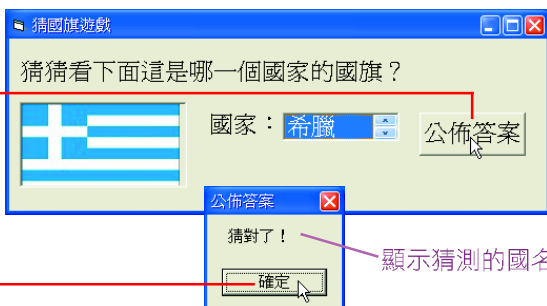
由於 ListBox 控制物件的註標值是由 0 開始，因此必須加 1，使其與亂數值的範圍一致，皆介於 1~7 之間

s.3 設定寬度為 190、高度為 210

3.1 按開始鈕 執行程式

3.2 選按希臘選項，並按公佈答案鈕，查看猜測的國家是否正確

3.3 按確定鈕，關閉此訊息交談窗



顯示猜測的國名正確

馬上練習

一、實作題

- 請開啓已佈妥控制物件的範例檔案 "Ex06-b.vbp"，利用 If-Then-Else 判斷敘述設計一個可計算身體質量指數 (BMI) 並判斷體重是否過重的程式，最後將檔案儲存為 "Sp06-b.vbp"。

提示：(1) 身體質量指數的計算公式： $BMI = \text{體重} / (\text{身高})^2$ ，其中體重的單位為公斤，身高的單位為公尺。

- 利用 If-Then-Else 判斷敘述限制使用者輸入的身高不能超過 2.5 公尺，體重不能超過 300 公斤。
- 男性與女性的 BMI 標準值分別為 27.8 及 27.3，若超過標準值即表示體重過重。

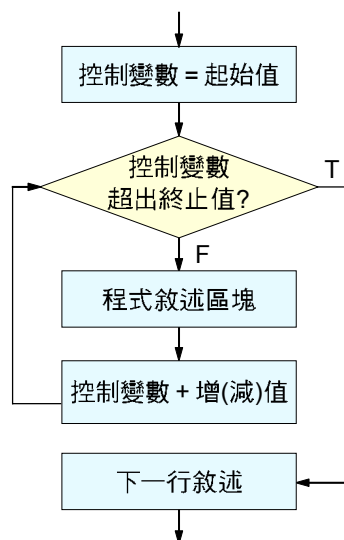


12-2.4 重覆結構

重覆結構是一種利用迴圈敘述來重複執行程式敘述的程式結構。下面我們將分別說明 Visual Basic 中常用的 For-Next 迴圈敘述、While-Wend 迴圈敘述、與 Do-Loop 迴圈敘述。

For-Next 迴圈敘述

當程式中的某段敘述必須重複執行，或已確定須執行的次數時，通常我們會使用 For-Next 迴圈敘述來處理。For-Next 迴圈敘述的執行流程若以流程圖表示，可繪製如圖 12-33 所示，其語法說明如下：



▲ 圖 12-33 For-Next 迴圈敘述的流程圖

```
Dim A(5) as Integer
Dim i as Integer
For i = 1 To 5
    A(i) = i * 10
Next i
```

▲ 圖12-34 For-Next迴圈敘述的實例

語法：

For 控制變數 = 起始值 To 終止值 Step 增值（減值）
程式敘述區塊

Next 控制變數

說明：

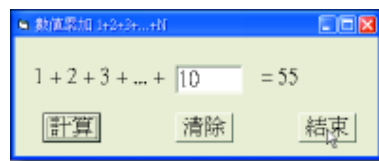
1. 第1次進入 For-Next 迴圈時，控制變數會被設為起始值。
2. 每次執行 For-Next 迴圈時，程式都會判斷控制變數是否超過終止值，如未超過，則繼續執行迴圈內的敘述，如超過，則跳出迴圈。
3. 當執行到「Next 控制變數」時，程式會將目前控制變數的值加上增值（或減值）。
4. 當增值為1時，Step 可省略不寫。

圖 12-34 為使用 For-Next 迴圈敘述的程式片段實例，此實例會設定陣列 A() 第 1 個到第 5 個資料元素的值。

程式設計實例－For-Next迴圈敘述

步驟 1：定義問題

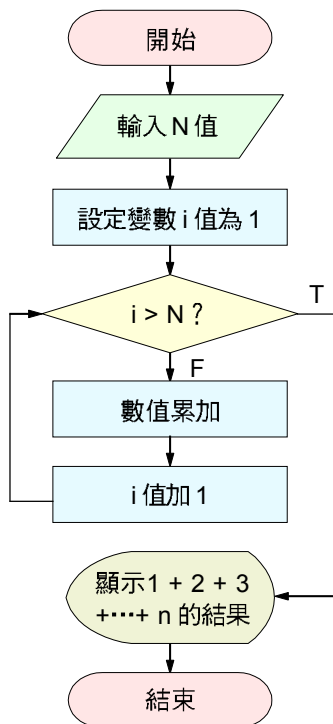
讓使用者輸入 N 值，然後計算及顯示 $1 + 2 + 3 + \dots + N$ 的總和（如圖 12-35）。



▲ 圖12-35 「數值累加」程式之執行結果

步驟 2：擬定解題計畫

- (1) 當使用者按**計算**鈕時，利用 For-Next 迴圈敘述計算 $1 + 2 + 3 + \dots + N$ 的總和，並顯示加總結果。
- (2) 當使用者按**清除**鈕時，清除在文字方塊中所輸入的資料。
- (3) 當使用者按**結束**鈕時，結束程式的執行。
- (4) 本實例之解題步驟，若以流程圖表示，可以繪製如圖 12-36 所示。



▲ 圖12-36 數值累加之流程圖

步驟 3：撰寫程式

請同學開啓已佈建控制物件的範例檔案 "Ex06-2d.vbp" (如圖 12-37 所示)，並依照操作實例 5 的說明，撰寫「數值累加」的程式。



▲ 圖12-37 範例檔案Ex06-2d.vbp

操作實例 5

「數值累加」程式設計

撰寫一個數值累加的程式，然後測試程式執行的結果。

```

Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
10 Dim N As Integer, i As Integer, sum As Long
20 N = Text1.Text
30 For i = 1 To N '計算1加到N的總和
40 sum = sum + i
50 Next i
60 Label2.Caption = "=" & sum
End Sub
  
```

30~50 敘述區塊為重覆結構

```

Project1 - Form1 (程式碼)
Command2 Click
Private Sub Command2_Click()
10 Text1.Text = ""
20 Label2.Caption = ""
30 Text1.SetFocus '將滑鼠游標移到文字方塊中
End Sub
  
```

SetFocus 是一種 TextBox 控制物件的方法，它可以將鍵盤輸入的游標移至文字方塊上

Step 1 在Command1的Click事件程序中撰寫計算 1 加到 N 之總和的程式碼

1.1 輸入程式碼 10~20，宣告變數，並設定 N 值

1.2 輸入程式碼 30~60，計算 1 加到 N 的總和，並將結果顯示在 Label2 中

Step 2 在Command2的Click事件程序中撰寫清除文字方塊內之資料的程式碼

2.1 輸入程式碼 10，清除文字方塊內的資料

2.2 輸入程式碼 20，將 Label2 控制物件的 Caption 屬性值設定為 "="

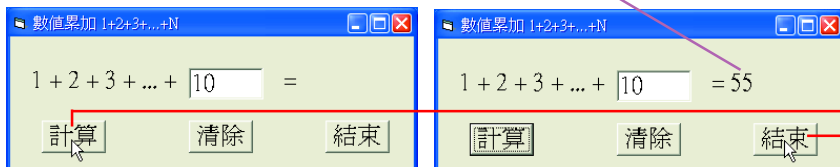
2.3 輸入程式碼 30，設定滑鼠游標出現在文字方塊中

Step 3 執行程式，測試程式執行的結果

3.1 按開始鈕 ▶ 執行程式

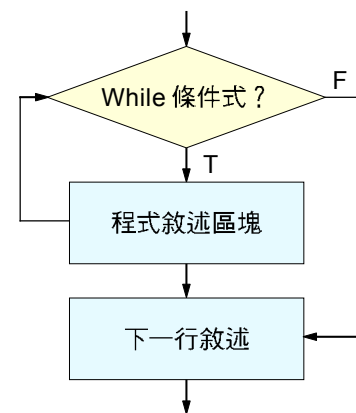
3.2 輸入 "10" 並按計算鈕

3.3 按結束鈕，結束程式的執行



While-Wend迴圈敘述

在撰寫須重複執行的程式敘述區塊時，如果不確定迴圈要執行的次數，可以使用 While-Wend 迴圈敘述來處理。While-Wend 迴圈敘述的執行流程若以流程圖表示，可繪製如圖 12-38 所示，其語法如下：



▲ 圖12-38 While-Wend迴圈敘述的流程圖

語法：

```
While 條件式
    程式敘述區塊
Wend
```

說明：

1. 若「條件式」成立，則執行迴圈中的程式敘述區塊。
2. 每次執行迴圈時，都必須重新判斷「條件式」是否成立。

圖 12-39 為使用 While-Wend 迴圈敘述來找出 30 的因數之程式實例。

！小提示

在撰寫程式碼時，冒號(:)可用來連接多行不同的程式敘述，將較短的程式敘述撰寫在同一行，以使程式較為簡潔。

```
i = 1: n = 30
Print "找出" & n & "的因數"
While i <= n
    If n Mod i = 0 Then Print i
    i = i + 1
Wend
```

▲ 圖12-39 While-Wend迴圈敘述的實例

程式設計實例— While-Wend 迴圈敘述

◎ 腦力激盪

如果要使用 While-Wend 迴圈敘述來計算 1 ~ 10 的奇數和，則程式敘述該怎麼撰寫呢？

步驟 1：定義問題

模擬彩券猜獎遊戲，讓使用者猜測電腦隨機產生的 6 個亂數，然後判斷使用者猜中了幾個亂數（如圖 12-40）。



▲ 圖 12-40 「公益彩券猜獎遊戲」程式之執行結果

步驟 2：擬定解題計畫



- (1) 當使用者按下**開獎**鈕時，程式隨機產生 6 個介於 1~42 之間的亂數。
- (2) 利用 While-Wend 迴圈敘述及 2 個各可儲存 6 個資料元素的陣列，儲存亂數值和使用者的數字。
- (3) 結合巢狀 For-Next 迴圈敘述及 If-Then 判斷敘述來判斷使用者猜中幾個亂數值。
- (4) 利用 For-Next 迴圈敘述及控制物件陣列（control array）來顯示代表亂數值的樂透彩球圖片。表 12-5 為控制物件陣列的用法說明。

表 12-5 控制物件陣列的用途與建立方法

簡介說明 控制物件陣列是利用複製物件的功能，複製出一組屬性相同的控制物件組，以簡化佈建多個相同類型之控制物件的時間，並讓程式設計人員可使用陣列名稱來引用這組控制物件。

控制物件陣列中的每一個控制物件都有一個唯一的索引值（功用同於陣列的註標）；程式設計人員可以利用索引值來識別陣列中各個不同的控制物件。

建立方法 建立控制物件陣列，可依循下面的步驟來完成建立控制物件陣列的工作：

1. 選取所要建立控制物件陣列的控制物件
2. 按一般工具列上的複製鈕 ，再按貼上鈕 
3. 在下圖之交談窗中，按是鈕，表示要建立控制物件陣列



4. 重複按貼上鈕，即可建立所要佈建的控制物件個數

- (5) 利用 If-Then-Else 敘述來顯示猜中與否的訊息。

小百科

巢狀 For-Next 迴圈敘述是指 For-Next 迴圈敘述中還有 For-Next 迴圈敘述。例如下圖所示之程式片段即是一個利用巢狀 For-Next 迴圈敘述來顯示九九乘法表的程式實例。

```
For i = 1 To 9
  For j = 1 To 9
    Print i & " x " & j & " = " & i*j;
  Next j
  Print ' 換行
Next i
```

腦力激盪

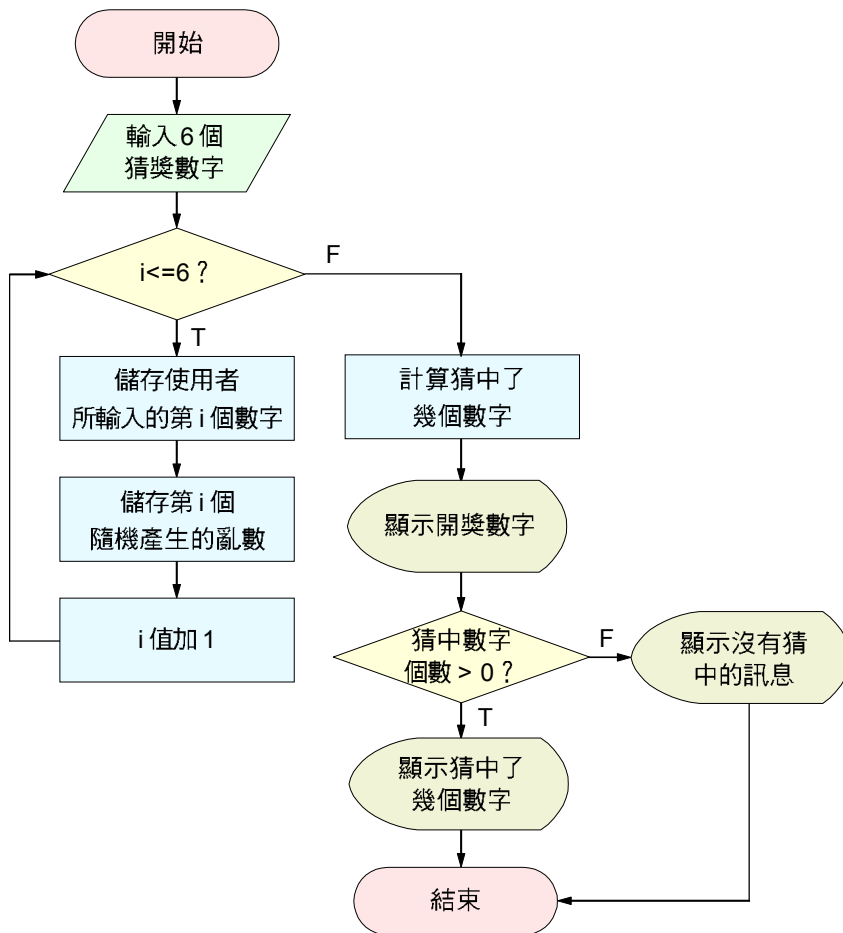
在 Visual Basic 中建立一個新專案，練習在表單上佈建 3 個標題控制物件。

(6) 本實例之解題步驟，若以流程圖表示，可繪製如圖 12-41 所示。

腦力激盪

下列程式執行後，表單上會顯示什麼結果？

```
S = 0
X = 1
While X < 50
  X = X + 1
  S = S + X
Wend
Print S
```



▲ 圖12-41 公益彩券猜獎遊戲之流程圖



▲ 圖12-42 範例檔案Ex06-2e.vbp

步驟3：撰寫程式

請同學開啓已佈建控制物件的範例檔案 "Ex06-2e.vbp" (如圖 12-42 所示)，並依照操作實例 6 的說明，撰寫「彩券猜獎遊戲」的程式。

操作實例 6 「彩券猜獎遊戲」程式設計

撰寫一個「彩券猜獎遊戲」的程式，然後測試程式執行的結果。

```

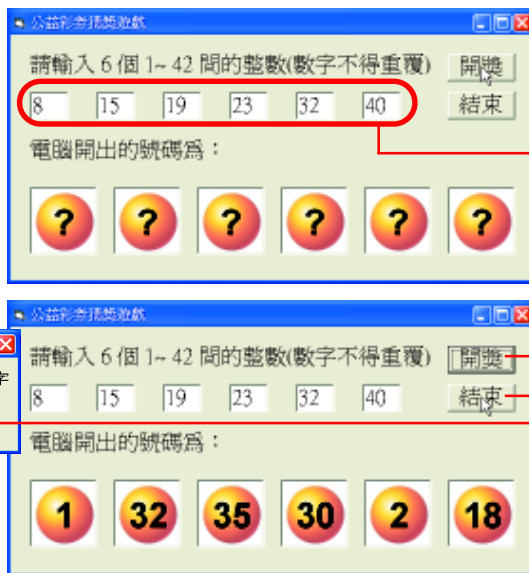
Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
10 Dim i As Integer, j As Integer, k As Integer
20 Dim a(6) As Integer, b(6) As Integer
30 i = 1: k = 0
40 While i <= 6 '儲存使用者猜測的數字及亂數值
50 a(i) = Text1(i - 1)
60 b(i) = Int(Rnd() * 42) + 1
70 i = i + 1
80 Wend
90 For i = 1 To 6 '判斷猜中幾個亂數值
100 For j = 1 To 6
110 If a(i) = b(j) Then
120 k = k + 1
130 End If
140 Next j
150 Next i
160 For i = 1 To 6 '載入代表亂數值的樂透彩球圖片
170 Picture1(i - 1).Picture = LoadPicture(App.Path & "樂透球"
180 & b(i) & ".jpg")
190 Next i
200 If k > 0 Then
210 MsgBox "您猜對了" & Str(k) & "個數字", "公益彩券"
220 Else
230 MsgBox "真可惜,沒有猜對!", "公益彩券"
240 End If
End Sub
    
```

Str(k) 可以將整數資料型別的變數 k，轉換成字串資料型別

當程式敘述太長而無法撰寫於一行時，則可使用底線（_）來接續程式敘述

Step 1 在Command1的Click事件程序中撰寫隨機產生亂數及判斷猜中幾個亂數的程式碼

- 1.1 輸入程式碼 10~30，宣告變數並設定變數的初始值
- 1.2 輸入程式碼 40~80，儲存使用者輸入的數字及電腦產生的亂數
- 1.3 輸入程式碼 90~140 判斷使用者猜中幾個亂數
- 1.4 輸入程式碼 150~180 載入代表亂數值的樂透彩球圖片
- 1.5 輸入程式碼 190~230，判斷使用者是否猜中並顯示相關的訊息



顯示猜中了幾個數字

Step 2 在Command1的Click事件程序中撰寫計算 1 加到 N 之總和的程式碼

- 2.1 按開始鈕，執行程式
- 2.2 依序輸入 6 個介於 1~42 之間的整數
- 2.3 按開獎鈕
- 2.4 按確定鈕，關閉此交談窗
- 2.5 按結束鈕，結束程式的執行

本範例檔案已在一般的宣告區中設定陣列註標下限值為 1。

Do-Loop 迴圈敘述

當程式中的某段敘述必須重複執行，且無法事先確定執行次數時，除了可以使用 While-Wend 迴圈敘述之外，也可以使用 Do-Loop 迴圈敘述來處理。Do-Loop 迴圈敘述可再細分為 **Do-Loop 前測式迴圈敘述**及 **Do-Loop 後測式迴圈敘述**兩種，分別說明如下：

！小提示

使用 "Exit Do" 程式敘述可強制跳離 Do-Loop 迴圈。

● **Do-Loop 前測式迴圈敘述**：所謂前測式迴圈是指先檢測迴圈之執行條件，再根據檢測結果決定是否執行的迴圈。Do-Loop 前測式迴圈有 Do-While-Loop 及 Do-Until-Loop 兩種，其語法說明如下：

語法 1： Do While 條件式
程式敘述區塊
Loop

語法 2： Do Until 條件式
程式敘述區塊
Loop

說明：

1. 語法 1 為 Do-While-Loop 前測式迴圈敘述，當 While 條件式成立時，執行迴圈內之程式敘述區塊；否則就跳出迴圈。
2. 語法 2 為 Do-Until-Loop 前測式迴圈敘述，當 Until 條件式成立時，跳出迴圈；否則就執行迴圈內之程式敘述區塊。

```
i = 2
Print "1到10中的偶數有 ";
Do While i <= 10
    Print i;
    i = i + 2
Loop
```

▲ 圖12-43 Do-Loop前測式迴圈敘述的實例

圖 12-43 為使用 Do-Loop 前測式迴圈敘述來輸出 1 到 10 之間的所有偶數的程式實例。

- **Do-Loop 後測式迴圈敘述**：所謂後測試迴圈是指先執行迴圈內的敘述一次，再檢測迴圈條件，以決定是否繼續執行迴圈。Do-Loop 後測試迴圈有 Do-Loop-While 及 Do-Loop-Until 兩種，其語法說明如下所示。

語法 1：Do

程式敘述區塊
Loop While 條件式

語法 2：Do

程式敘述區塊
Loop Until 條件式

說明：

1. 語法 1 為 Do-Loop-While 後測式迴圈敘述，執行 Do 迴圈程式敘述區塊後，當 While 條件式成立時，將繼續執行迴圈；否則就跳出迴圈。
2. 語法 2 為 Do-Loop-Until 後測式迴圈敘述，執行 Do 迴圈程式敘述區塊後，當 Until 條件式成立時，將跳出迴圈；否則就繼續執行迴圈。

小百科

無窮迴圈 (endless loop) 是一種因迴圈之條件式設計不當，而使得程式在執行時，無法跳離迴圈內之程式敘述的迴圈。

```
i = 1
Print "1到10中的奇數有";
Do
    Print i;
    i = i + 2
Loop While i <= 10
```

圖 12-44 為使用 Do-Loop 後測式迴圈敘述來輸出 1 到 10 之間的所有奇數的程式實例。

▲ 圖12-44 Do-Loop後測式迴圈敘述的實例

知識頻道**無條件跳躍的 GoTo 敘述**

GoTo 敘述可以用來無條件跳到指定的程式位置來執行，其語法及實例如圖 12-45 所示。在程式設計時，使用 GoTo 敘述將會強制改變程式執行的流程，因而使程式執行順序的追蹤及日後的維護都較為困難；因此在撰寫程式時，應盡量避免使用此種敘述。

用來標示程式中
某一敘述的位置

```
GoTo 標記
...
標記:程式敘述
```

GoTo 的語法

```
a = 1
line:Print a
a = a + 1
if a <=10 Then GoTo line
Print "end"
```

GoTo 的應用實例

▲ 圖12-45 GoTo的語法與應用實例

馬上練習

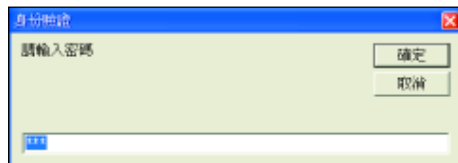
一、實作題

1. 利用 For-Next 迴圈敘述（或 While-Wend 迴圈敘述、Do-Loop 迴圈敘述）及 If-Then-Else 敘述設計一個可讓使用者輸入密碼的程式，最後將檔案儲存為 "Sp06-c.vbp"。

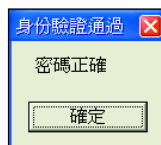
提示：(1) 利用 InputBox() 函數讀取使用者輸入之密碼（假設正確的密碼是 "op5e4"）。

- (2) 密碼輸入錯誤時，程式須顯示警告訊息；若連續3次輸入之密碼皆錯誤，程式便會立即中止執行。

- (3) 利用 For-Next 迴圈敘述來控制使用者輸入密碼的次數。若輸入之密碼正確，利用 "Exit For" 敘述來強迫跳離 For-Next 迴圈。



密碼正確



密碼錯誤



12-2 節練習

一、選擇題

- @ () 1. 執行下列 Visual Basic 程式片段，程式輸出的結果為何？ (A)8 (B)5 (C)2 (D)1。

```
A = 5: B = 2: C = 1
```

```
If A > B Then
```

```
    C = A
```

```
Else
```

```
    C = B
```

```
End If
```

```
Print C
```

- @ () 2. 執行下列 Visual Basic 程式，程式輸出的結果為何？ (A)error (B)large (C)medium (D)small。

```
Private Sub Form_Activate()
```

```
    X = 5 \ 2
```

```
    Select Case X
```

```
        Case 1: Print "small"
```

```
        Case 2: Print "medium"
```

```
        Case Else: Print "large"
```

```
    End Select
```

```
End Sub
```

- @ () 3. 執行下列 Visual Basic 程式片段，螢幕上會輸出幾個 "#" 號？ (A)6 (B)7 (C)8 (D)9。

```
For i = 19 To 1 Step -3
    Print "#";
Next i
```

- @ () 4. 執行下列 Visual Basic 程式片段，若在輸入資料的訊息交談窗中輸入數值 "100"，則程式的執行結果為何？ (A)5 (B)6 (C)64 (D)128。

```
n = 0
a = InputBox("input data")
While a > 0
    a = a - 2 ^ n
    n = n + 1
Wend
Print n - 1
```

- @ () 5. 執行下列 Visual Basic 程式片段，程式輸出的結果為何？ (A)D = 1680 (B)D = 168000 (C)D = 15120 (D)D = 151200。

```
C = 5: D = 1
Do
    D = D * C
    C = C + 1
Loop Until C >= 10
Print "D = "; D
```

- @ () 6. 執行下列 Visual Basic 程式片段，A(1, 3) 的值與下列哪一個元素的值相同？ (A)A(2, 0) (B)A(2, 1) (C)A(2, 2) (D)A(2, 3)。

```
Dim A(3, 3) As Integer
For I = 0 To 3
    For J = 0 To 3
        A(I, J) = 2 * I * (I + J)
    Next J
Next I
```

二、實作題

1. 本節「彩券猜獎遊戲」程式 (P.306) 在執行時，有可能會發生開出相同號碼的問題，請參考以下的提示修改程式，以解決此問題。

提示：(1) 利用一維陣列來記錄開獎號碼。

(2) 使用條件判斷及迴圈敘述來判斷開獎號碼是否重覆。

12-3 結構化程式設計

小百科

結構化程式設計（structured programming）是由軟體工程專家Edsger Wybe Dijkstra所提倡的一種設計程式的觀念及撰寫程式的方法。

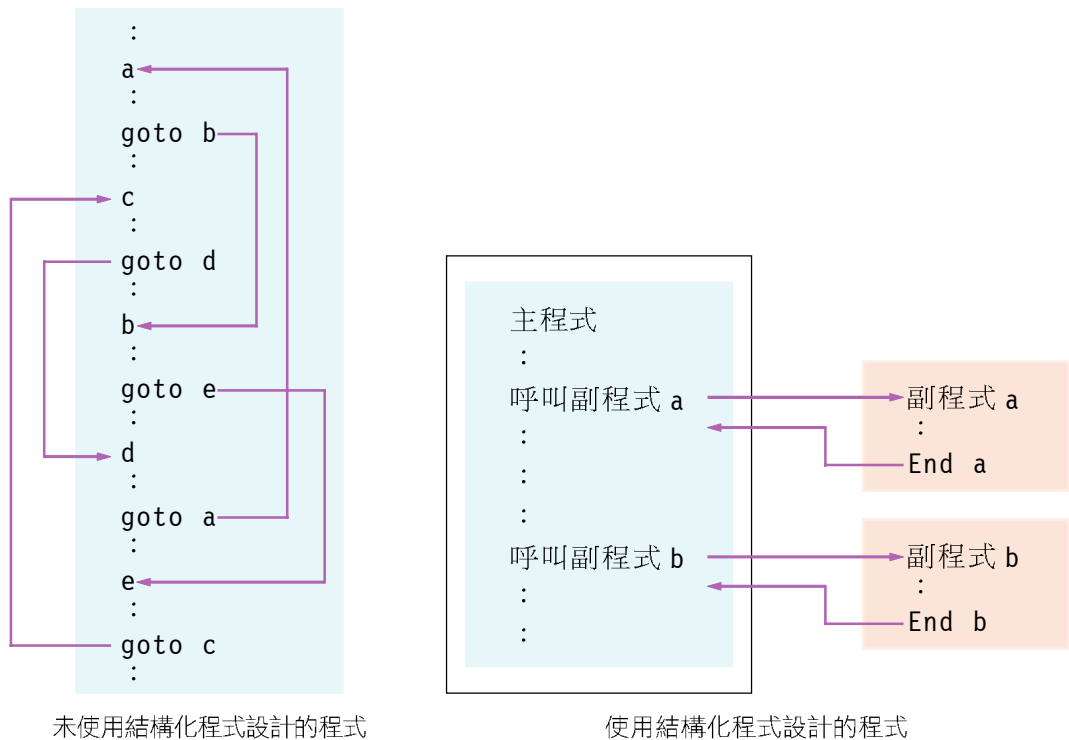
相關連結

<http://134.208.27.113/chap3.ppt>
瞭解結構化程式開發

http://noi.stinfo.net/xbqj/xbqj_12.htm
閱讀結構化程式設計的方法

程式設計是一種發揮個人創意來設計程式的心思活動。在軟體程式發展早期，由於程式設計人員常強調個人的設計風格，再加上隨意使用 GOTO 敘述來改變程式的執行流程，因此所設計的程式常有不易閱讀、不易除錯、及維護困難等缺點。

爲了改善上述程式設計的缺點，電腦學者便提出了**結構化程式設計**（structured programming）的概念來改善程式設計工作的效率與品質。圖 12-46 是使用結構化程式設計及未使用結構化程式設計之程式的比較。本節將介紹結構化程式設計的概念，以及結構化程式設計常需使用到的副程式與函數。



▲ 圖12-46 使用結構化程式設計及未使用結構化設計之程式比較

12-3.1 結構化程式設計的概念

結構化程式設計是一種可使程式的設計與維護較易於進行的程式設計技術。這種程式設計技術通常遵循由上到下的程式設計（top-down programming）、模組化的程式設計（modulized programming）、及使用程式基本結構等三個原則來進行程式設計的工作。下面將介紹這三個結構化程式設計的原則。

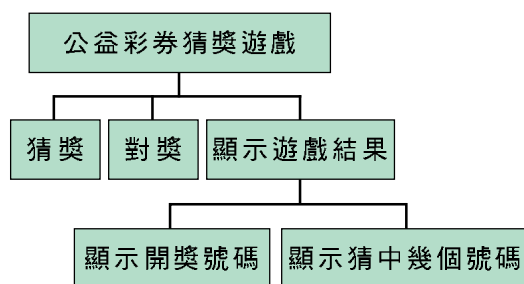
由上到下的程式設計

由上到下的程式設計是指在進行程式設計時，先由整個程式的主要功能開始設計，然後再依序往下設計各個子功能，直到最底層的功能設計完成為止。

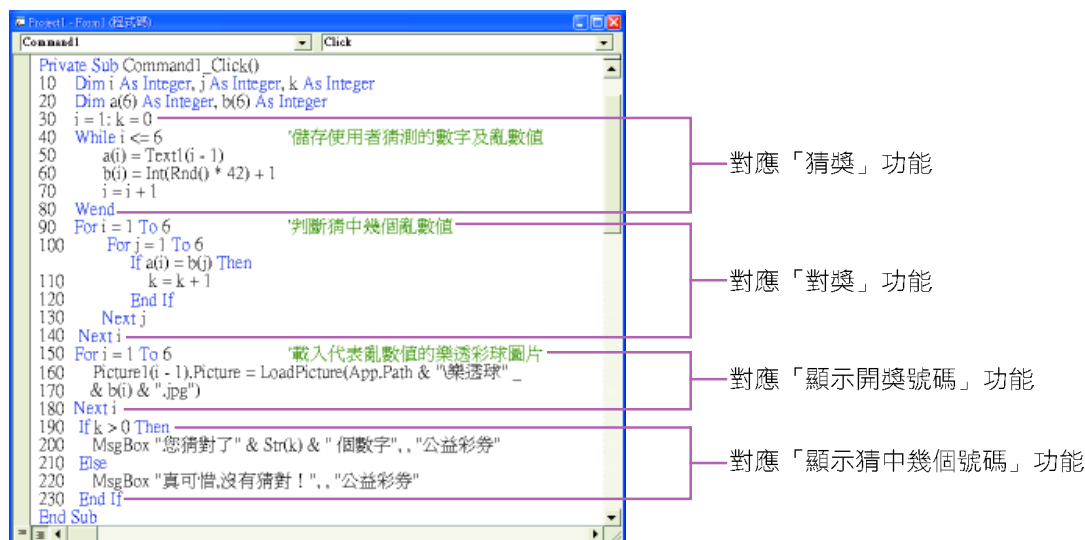
圖 12-47 為利用「由上到下的程式設計」概念，將 12-2 節公益彩券猜獎遊戲的程式設計實例，由上到下細分為「猜獎」、「對獎」、「顯示遊戲結果」、「顯示開獎號碼」、及「顯示猜中幾個號碼」等功能。圖 12-48 為各個功能與程式碼的對應關係。

一點就通

班上辦園遊會時，老師會先將同學分成佈置攤位、採購、販售…等組別，每組再由組長分配組員所需負責的工作，這就類似一個簡單的由上到下分派工作的設計方法。



▲ 圖 12-47 「公益彩券猜獎遊戲」的各個子功能



▲ 圖 12-48 「公益彩券猜獎遊戲」的各個功能與程式碼對應關係

模組化的程式設計

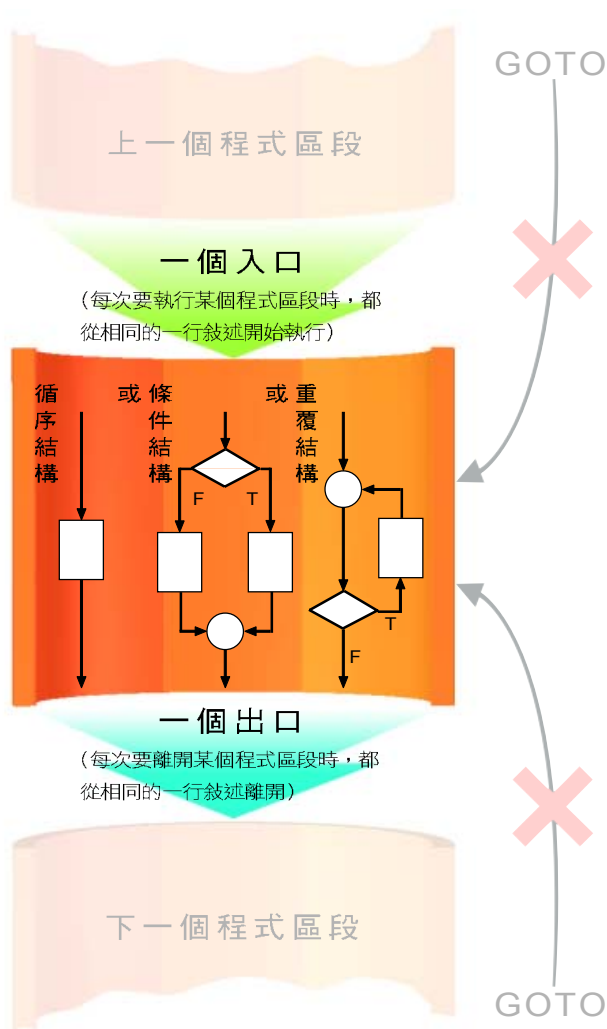


▲ 圖 12-49 由多個模組可組合成一個大型的程式

模組（module）通常是指一段具有某種特定功能、大小適中、容易閱讀及維護的程式。當我們遵循「由上到下的程式設計」原則，將程式切割成許多個子功能後，便可將這些子功能分別撰寫成模組程式（圖 12-49），例如我們可將圖 12-47 最底層的子功能，分別設計成模組程式。

爲了讓模組可供更多的程式使用，且能使模組易於測試及維護，在設計模組時應儘量依循以下三點原則：

- 設計功能單純的模組程式，以便提供給更多的程式使用。
- 儘量減少模組之間相互影響的程度，以使模組成爲獨立的程式基本單元。
- 將模組控制在適當的大小（例如不超過 50 行敘述），以方便閱讀與維護。



使用程式基本結構

循序、選擇、及重覆是程式的三種基本結構，在撰寫程式時，應儘量使用這三種程式的基本結構，並避免使用 GOTO 敘述來改變程式的執行流程，讓每一個程式區段都只有一個入口與出口，如圖 12-50 所示。

▲ 圖 12-50 遵守一個入口與出口的程序設計原則

馬上練習

一、選擇題

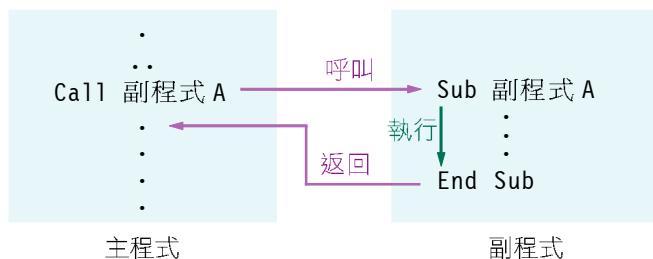
- @ () 1. 下列哪三個程式基本結構是結構化程式設計常用的控制結構？(A)循序、索引、選擇 (B)循序、選擇、重覆 (C)索引、選擇、重覆 (D)樹狀、循序、重覆。
- @ () 2. 下列何者不是結構化程式設計的原則（方法）？(A)由上到下的設計方式 (B)重覆撰寫功能相同的程式敘述以利於閱讀 (C)避免使用GOTO敘述 (D)採用模組化的設計技術。
- @ () 3. 下列何者不是結構化程式的優點？(A)容易了解 (B)容易維護 (C)容易測試及除錯 (D)減少CPU執行的時間。

12-3.2 結構化程式設計的實作

模組化是結構化程式設計的重要方法；在 Visual Basic 中，我們可以使用**副程式**（subprogram）及**函數**（function）將大型的程式分割成許多個功能單純的小型模組，以利程式撰寫、測試、及維護。下面我們將分別說明副程式及函數的使用方法，並以實例來介紹結構化程式設計的技巧。

副程式

副程式是一段獨立的程式區塊，由 Sub 與 End Sub 之間的程式敘述所組成。每個副程式都有一個專屬的名稱；當程式中的其它區塊需要使用這個副程式的功能時，可以使用**呼叫**（call）的方式來執行這一段副程式（如圖 12-51 所示）。



▲ 圖12-51 主程式呼叫副程式的示意圖

副程式的語法說明

副程式的撰寫及呼叫之語法說明如下：

 **相關連結**
<http://www.socialwork.com.hk/vbstudy.htm>
 瀏覽副程式的撰寫實例

語法 1：

Sub 副程式名稱 (參數 1 As 資料型別, 參數 2 As 資料型別, …)
 程式敘述區塊

End Sub

語法 2：

Call 副程式名稱 (參數 1, 參數 2, …)

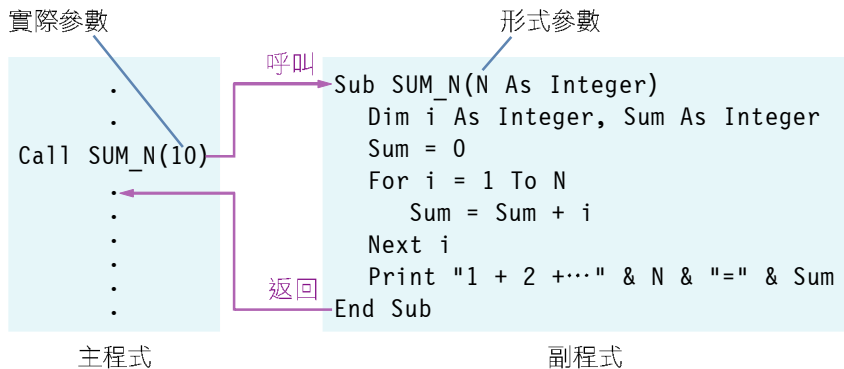
說明：

1. 語法 1 為設計副程式使用的語法；語法 2 為呼叫副程式使用的語法。
2. 在 Sub 敘述中可加入要傳送給副程式處理的**參數** (parameters)，並設定參數的資料型別。
3. 如果 Sub 敘述中的參數未宣告資料型別，則視為「可變資料型別」(variant)。
4. 在呼叫副程式時，可在 Call 敘述中將參數值傳給副程式，以便副程式處理。

小百科

在 Sub 敘述中加入的參數稱為形式參數；而在 Call 敘述中所代入的參數稱為實際參數，例如圖 12-51 所示的程式實例，形式參數為 N；實際參數為 10。

圖 12-52 為在主程式中呼叫副程式 SUM_N，並將參數值 10 傳給該副程式，以計算 1 加到 10 之總和的程式實例。



▲ 圖12-52 副程式的實例

副程式的參數傳遞

Visual Basic 提供兩種參數傳遞的方式：**傳址**（pass by reference）與**傳值**（pass by value），分別說明如下：

- **傳址**：所謂傳址的方式，是指在呼叫副程式（或函數）時，會將實際參數在記憶體中的位址傳遞給形式參數。當副程式（或函數）使用到形式參數時，便會由該位址取用實際參數的值來做運算。

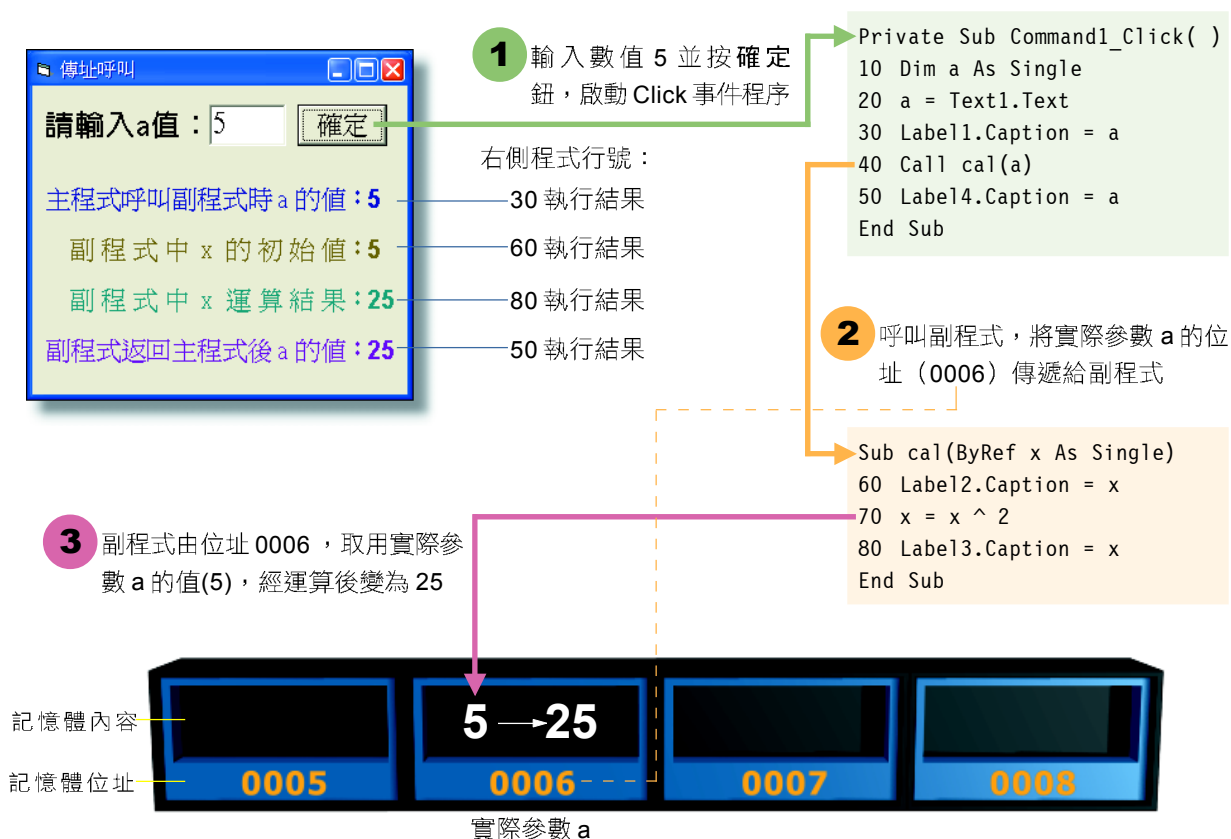
在 Visual Basic 中，若要以傳址的方式來呼叫副程式（或函數），可在副程式（或函數）中的形式參數前加上保留字 "ByRef" 或省略不寫。圖 12-53 為傳址呼叫的程式實例。

相關連結

<http://www.geog.ntu.edu.tw/course/vb/第二講.files/frame.htm> 閱讀 VB 副程式（或函數）傳遞參數的方法

小提示

Visual Basic 預設的參數傳遞方式為「傳址」。



▲ 圖12-53 傳址呼叫副程式之示意圖

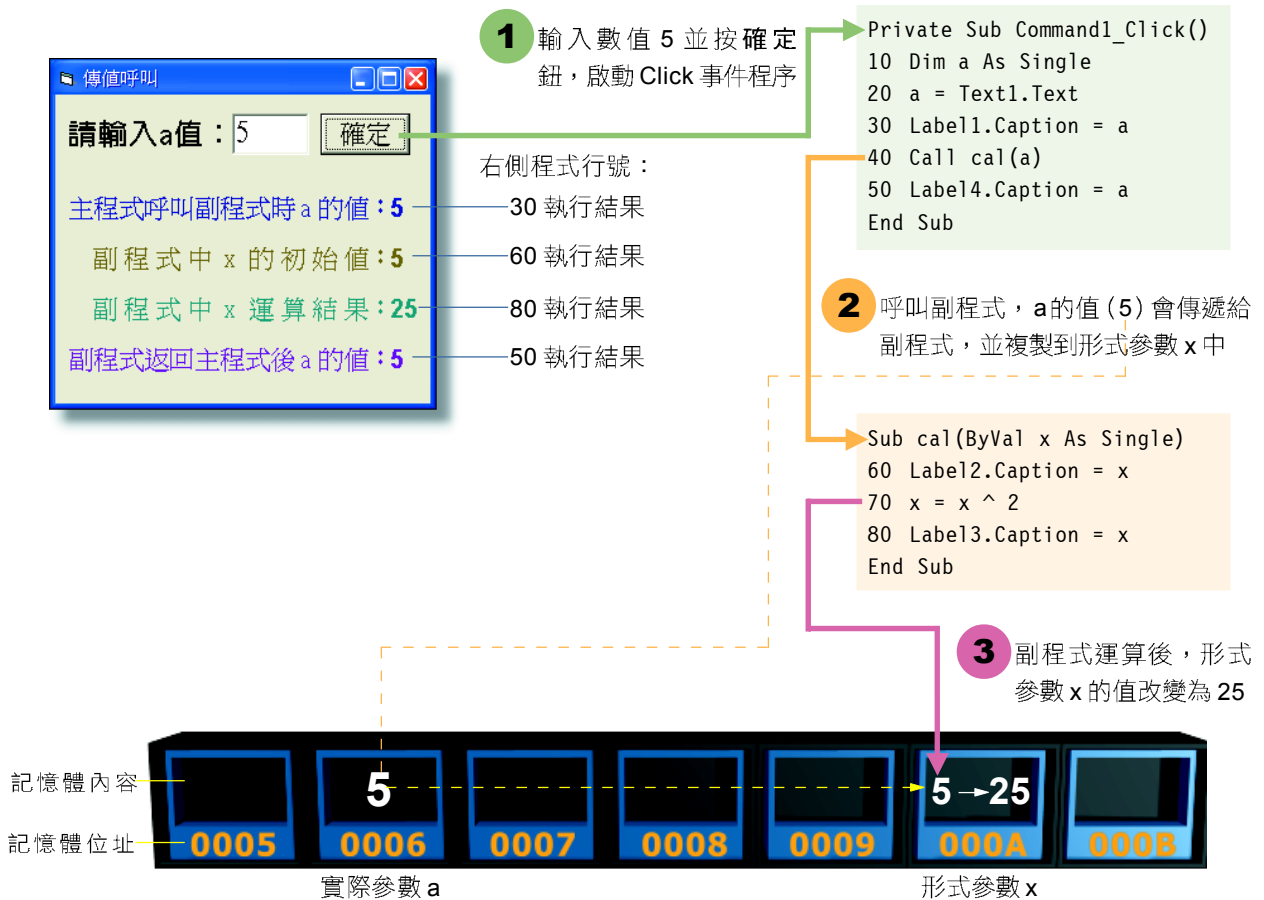
在以傳址方式呼叫副程式之後，實際參數 a 的值由 5 改變為 25，故視窗中所顯示的計算結果為 25。

！小提示

若要讓實際參數的值在呼叫副程式（或函數）後，隨著形式參數的運算而改變，可以使用傳址的方式來傳遞參數。反之，若要在呼叫副程式後，仍維持實際參數的值，則可使用傳值的方式來傳遞參數。

● **傳值**：所謂傳值的方式，是指在呼叫副程式（或函數）時，會將實際參數的值複製到形式參數中，並直接使用該複製的值來運算。

在 Visual Basic 中，若要以傳值的方式來呼叫副程式（或函數），可在副程式（或函數）中的形式參數前加上保留字 "ByVal"，或者在呼叫該副程式（或函數）敘述中的實際參數前後加上左右小括號，例如：Call cal((a))。圖 12-54 為傳值呼叫的程式實例。



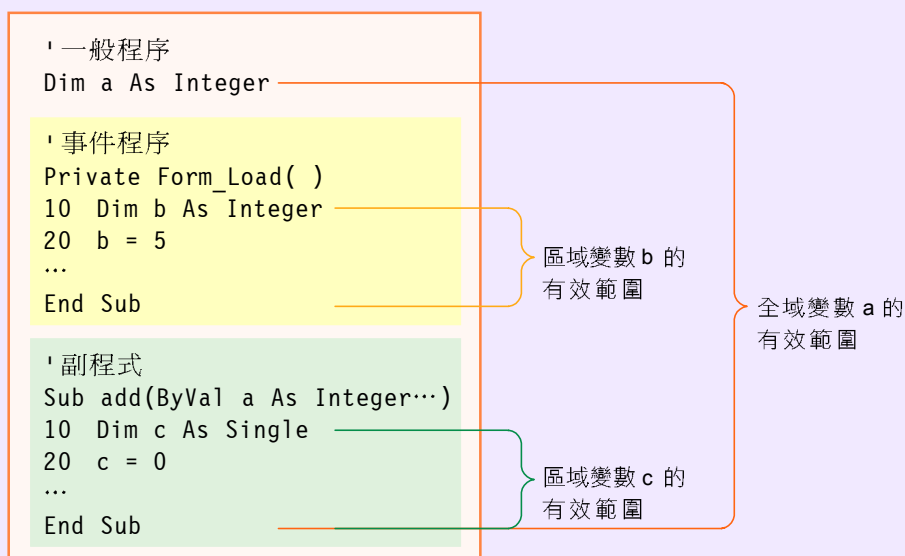
▲ 圖12-54 傳值呼叫副程式之示意圖

知識頻道

全域變數與區域變數

在 Visual Basic 中，一般程序內所宣告的變數，稱為全域變數（global variables），這種變數的有效範圍可遍及整個程式。而在事件程序、副程式（或函數）中宣告的變數，則稱為區域變數（local variables），這種變數的有效範圍僅限於宣告它的事件程序、副程式（或函數）內。圖 12-55 為全域變數及區域變數的有效範圍示意圖。

區域變數只有當事件程序、副程式（或函數）執行時才會存在，一旦事件程序或副程式執行完畢，該事件程序或副程式中的區域變數之生命週期即告結束，直到另一次的呼叫，區域變數才會再次產生。因此，區域變數的值在前後兩次呼叫事件程序或副程式之間是無法延續保存的。



▲ 圖12-55 全域變數及區域變數的有效範圍示意圖

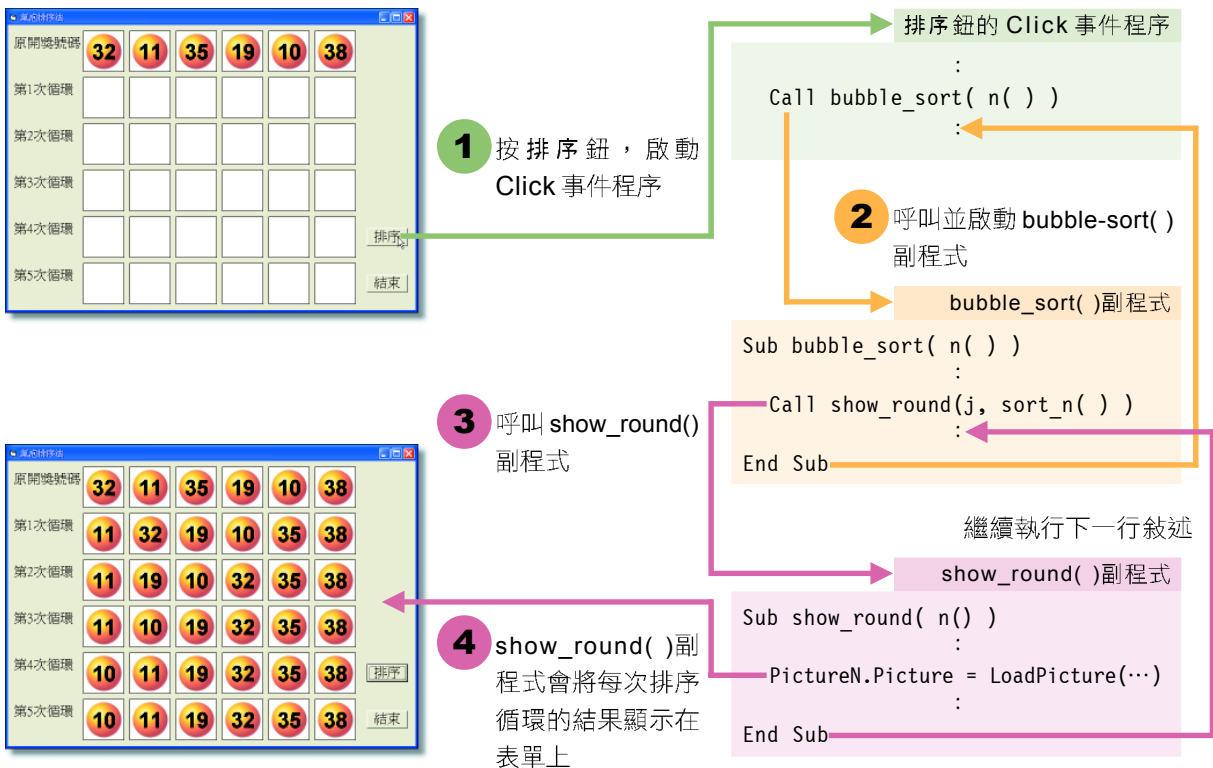
副程式的設計實例

步驟 1：定義問題

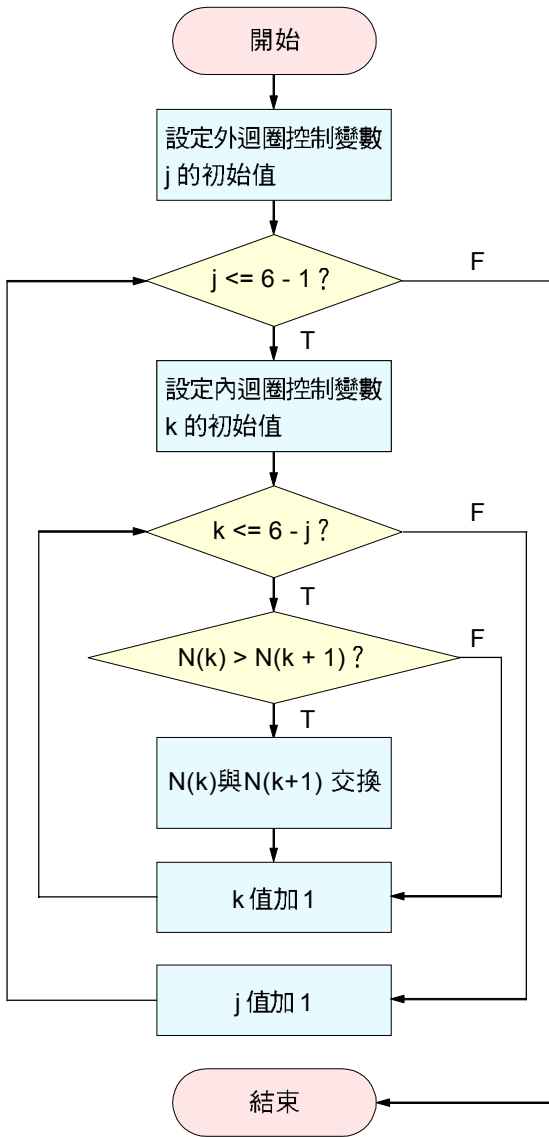
本例將延續 12-2 節公益彩券猜獎遊戲的程式設計實例，將 6 個隨機產生的數值以氣泡排序法由小到大排序。

步驟2：擬定解題計畫

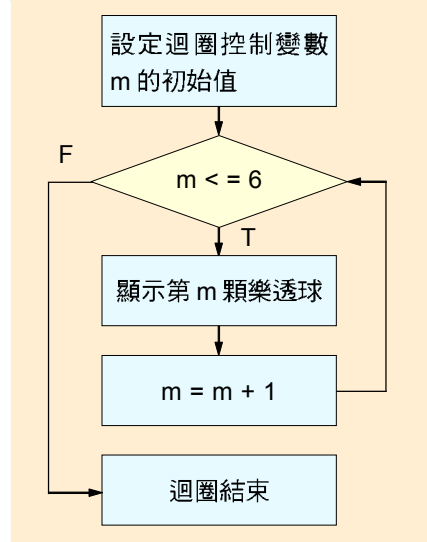
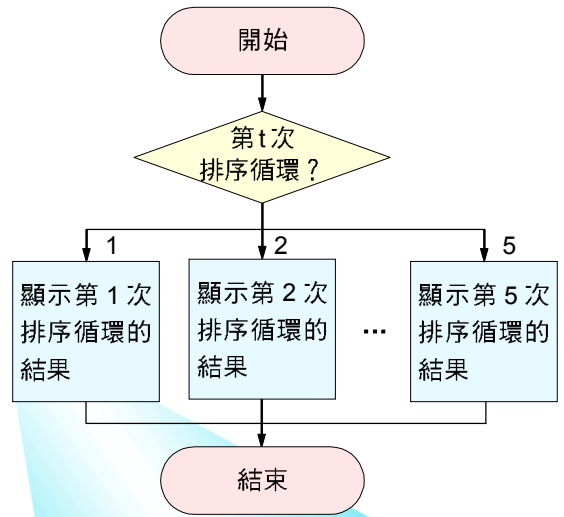
- (1) 表單載入時，隨機產生 6 個介於 1~42 之間的數值，並載入代表各數值的樂透彩球圖片。
- (2) 當使用者按**排序**鈕時，程式便呼叫 `bubble_sort()` 副程式進行數值的排序，副程式 `bubble_sort()` 每完成一次排序循環，便會呼叫副程式 `show_round()`，在表單上輸出排序的結果。
- (3) 圖 12-56 為本實例進行樂透彩球排序及顯示的執行過程示意圖；圖 12-57 及圖 12-58 分別為 `bubble_sort()` 與 `show_round()` 副程式的執行流程圖。



▲ 圖12-56 利用 `bubble_sort()` 與 `show_round()` 副程式進行樂透彩球排序及顯示的示意圖



▲ 圖 12-57 bubble_sort() 副程式進行樂透彩球排序的流程圖

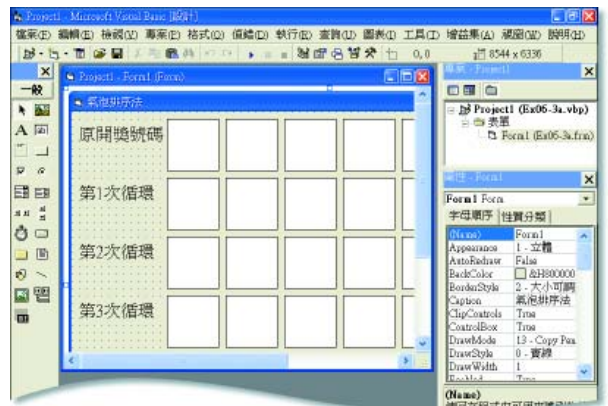


顯示第 n 次循環排序的流程

▲ 圖 12-58 show_round() 副程式顯示樂透彩球的流程圖

步驟 3：撰寫程式

請同學開啓已佈妥控制物件的範例檔案 "Ex06-3a.vbp" (如圖 12-59 所示)，並依照操作實例 7 的說明，撰寫「氣泡排序法」的程式。



▲ 圖 12-59 範例檔案 Ex06-3a.vbp

操作實例 7

「氣泡排序」程式設計

Step 1 在Form的Load事件程序中撰寫可隨機產生6個代表樂透彩球的程式碼

1.1 輸入程式碼 10，宣告變數 i 為整數資料型別

1.2 輸入程式碼 20，設定亂數的初始值

1.3 輸入程式碼 30~60，隨機產生6個介於1~42之間的亂數，並在 Picture1 控制物件陣列中載入樂透彩球圖片

利用「氣泡排序法」原理撰寫一個由小到大排序的程式，然後測試程式的執行結果。

Randomize 函數是 Visual Basic 所內建的數值型函數，它可用來設定亂數的序列，使程式每次執行時，都不會產生同一組的亂數

```

Project1 - Form1 (程式碼)
Form
Load
Private Sub Form_Load()
10 Dim i As Integer
20 Randomize '設定亂數的初始值
30 For i = 1 To 6
40     N(i) = Int(Rnd() * 42) + 1
50     Picture1(i).Picture = LoadPicture(App.Path & _
        "樂透球" & N(i) & ".jpg")
60 Next
End Sub
  
```

Step 2 在 Command1 的 Click 事件程序中撰寫呼叫副程式 bubble_sort()

2.1 輸入程式碼 10，呼叫副程式 bubble_sort()

```

Project1 - Form1 (程式碼)
Command1
Click
Private Sub Command1_Click()
10 Call bubble_sort(N())
End Sub
  
```

Step 3 撰寫 bubble-sort 副程式

3.1 輸入程式碼設定陣列註標下限由 1 開始，並宣告陣列變數 N

3.2 輸入程式碼設定副程式 bubble_sort() 的起始敘述

3.3 輸入程式碼 10 宣告變數

3.4 輸入程式碼 20~90，進行亂數值的比較

3.5 輸入程式碼 100~110，呼叫副程式 show_round()

當我們輸入副程式的起始敘述後，只要按下 **Enter** 鍵，程式碼視窗即會自動顯示副程式的結尾敘述(即 "End Sub")

```

Project1 - Form1 (程式碼)
(一般)
bubble_sort
Option Explicit
Option Base 1
Dim N(6) As Integer
Sub bubble_sort(N() As Integer)
10 Dim temp As Integer, j As Integer, k As Integer
20 For j = 1 To 6 - 1 '設定排序循環的次數
30     For k = 1 To 6 - j '依排序循環的次數增加，遞減比較次數
40         If N(k) > N(k + 1) Then '比較資料
50             temp = N(k) '交換資料
60             N(k) = N(k + 1)
70             N(k + 1) = temp
80         End If
90     Next k
100 Call show_round(j, N())
110 Next j
End Sub
  
```

```

Project1 - Form1 (程式碼)
(一般) show_round
Sub show_round(t As Integer, N() As Integer)
10 Dim m As Integer
20 Select Case t
    Case 1 '第1次循環
        For m = 1 To 6
            Picture2(m).Picture = LoadPicture(App.Path & _
                "\樂透球" & N(m) & ".jpg")
        Next m
    Case 2 '第2次循環
        For m = 1 To 6
            Picture3(m).Picture = LoadPicture(App.Path & _
                "\樂透球" & N(m) & ".jpg")
        Next m
    Case 3 '第3次循環
        For m = 1 To 6
            Picture4(m).Picture = LoadPicture(App.Path & _
                "\樂透球" & N(m) & ".jpg")
        Next m
    Case 4 '第4次循環
        For m = 1 To 6
            Picture5(m).Picture = LoadPicture(App.Path & _
                "\樂透球" & N(m) & ".jpg")
        Next m
    Case 5 '第5次循環
        For m = 1 To 6
            Picture6(m).Picture = LoadPicture(App.Path & _
                "\樂透球" & N(m) & ".jpg")
        Next m
30 End Select
End Sub

```

Step 4 撰寫 show_round 副程式

4.1 輸入程式碼，設定副程式 show_round() 的起始敘述

4.2 輸入程式碼 10，設定變數 m 為整數資料型別

4.3 輸入程式碼 20~30，依照參數值 t，在 Picture 控制物件陣列中載入第 t 次循環排序的結果

電腦產生的 6 個亂數值

原開獎號碼	25	32	18	10	7	13
第1次循環						
第2次循環						
第3次循環						
第4次循環						
第5次循環						

排序 結束

表單上顯示了 5 次循環的排序結果

原開獎號碼	25	32	18	10	7	13
第1次循環	25	18	10	7	13	32
第2次循環	18	10	7	13	25	32
第3次循環	10	7	13	18	25	32
第4次循環	7	10	13	18	25	32
第5次循環	7	10	13	18	25	32

排序 結束

Step 5 測試程式的執行結果

5.1 按開始鈕，執行程式

5.2 按排序鈕，以顯示氣泡排序法每次循環的排序結果

5.3 按結束鈕，結束程式的執行

函數

在 Visual Basic 中，函數分為**內建函數**（built-in function）及**自定函數**（user-defined function）兩種。內建函數是 Visual Basic 預先設計好的程式，使用者可直接輸入函數名稱來使用這些函數的功能；而自定函數則是使用者依照自己的需求，所自行撰寫的函數。

內建函數

Visual Basic 提供了相當豐富的內建函數，可供使用者直接使用，以省去自行撰寫程式的時間。表 12-6 列出了常用的字串型、數值型、及時間日期型三種內建函數。

腦力激盪

請利用即時運算視窗輸入以下程式，並檢視其執行結果。

- ```
a = "123"
b = "456"
c = Val(a) + Val(b)
Print c
```
- ```
x = 25
y = Sqr(x)
z = Right("Visual Basic", y)
Print z
```
- Print Month(Date())

腦力激盪

利用 Rnd() 函數，撰寫一個可模擬擲骰子所出現點數（1~6）的程式。

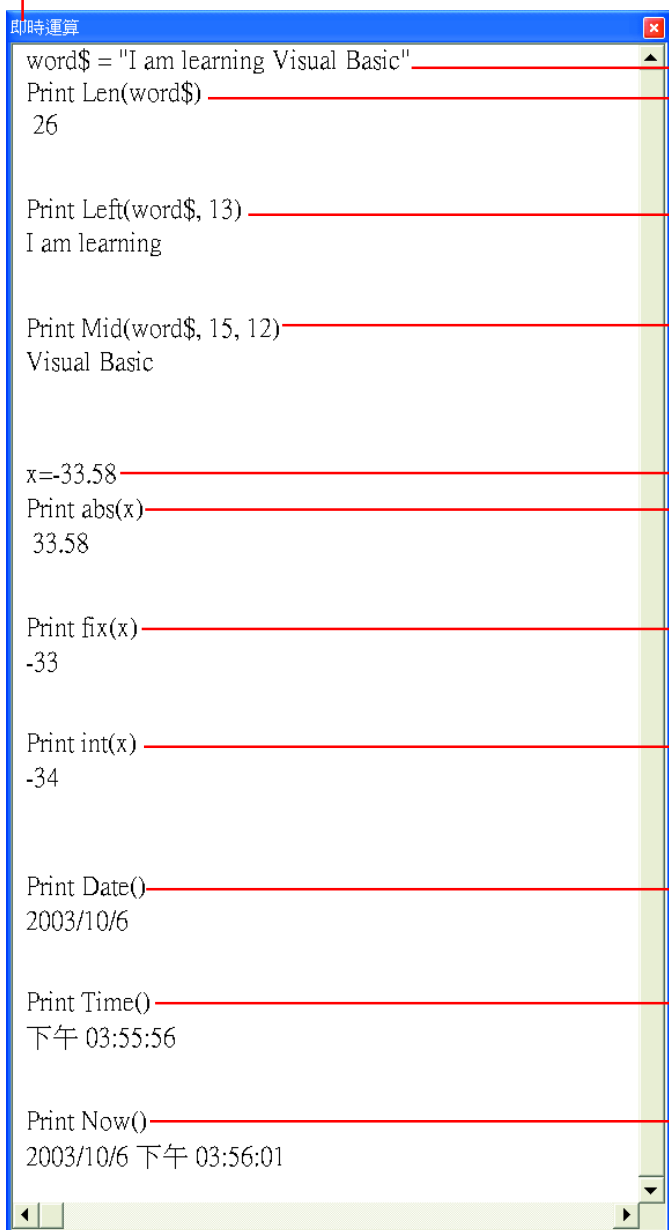
表 12-6 常用的 Visual Basic 內建函數及語法說明

類型	函數名稱與語法	說明	
字串型	Asc(字串)	傳回字串中第一個字元的 ASCII 碼	
	Len(字串)	計算字串的長度	
	Left(字串, N)	傳回字串由左算起 N 個字元	
	Right(字串, N)	傳回字串由右算起 N 個字元	
	Mid(字串, M, N)	傳回字串由第 M 個字元算起 N 個字元	
	Ltrim(字串)	去除字串開頭（左邊）的所有空白字元	
	Rtrim(字串)	去除字串結尾（右邊）的所有空白字元	
	Trim(字串)	去除字串頭尾的所有空白字元	
	Space(N)	空出 N 個空白字元	
	Str(數值)	將數值型別的資料轉換成字串型別的資料	
	數值型	Val(字串)	將字串型別的資料轉換成數值型別的資料
		Abs(數值)	取數值的絕對值
		Int(數值)	取小於或等於數值的最大整數值
Fix(數值)		取數值的整數部份，小數無條件捨去	
Sqr(數值)		取數值的平方根	
Rnd()		產生一個 ≥ 0 且 < 1 的亂數	
時間日期型		Date()	傳回系統的日期
	Time()	傳回系統的時間	
	Now()	傳回系統的時間與日期	
	Year(日期)	傳回日期資料中的年份資料	
	Month(日期)	傳回日期資料中的月份資料	

操作實例 8

熟悉 Visual Basic 常用的內建函數

開啟 Visual Basic 的即時運算視窗，練習內建函數的使用方法。



```
即時運算
word$ = "I am learning Visual Basic"
Print Len(word$)
26

Print Left(word$, 13)
I am learning

Print Mid(word$, 15, 12)
Visual Basic

x=-33.58
Print abs(x)
33.58

Print fix(x)
-33

Print int(x)
-34

Print Date()
2003/10/6

Print Time()
下午 03:55:56

Print Now()
2003/10/6 下午 03:56:01
```

1 選按『檢視 / 即時運算視窗』選項，開啟即時運算視窗

2 輸入程式碼，設定字串變數 word 的值

3 輸入程式碼，顯示字串變數 word 的長度

4 輸入程式碼，顯示字串變數 word 由左算起 1~13 個字元

5 輸入程式碼，顯示字串變數 word 由第 15 個字元算起的 12 個字元

6 輸入程式碼，設定變數 x 的值

7 輸入程式碼，顯示變數 x 的絕對值

8 輸入程式碼，顯示變數 x 的整數值

9 輸入程式碼，顯示變數 x 的最大整數值

10 輸入程式碼，顯示系統的日期

11 輸入程式碼，顯示系統的時間

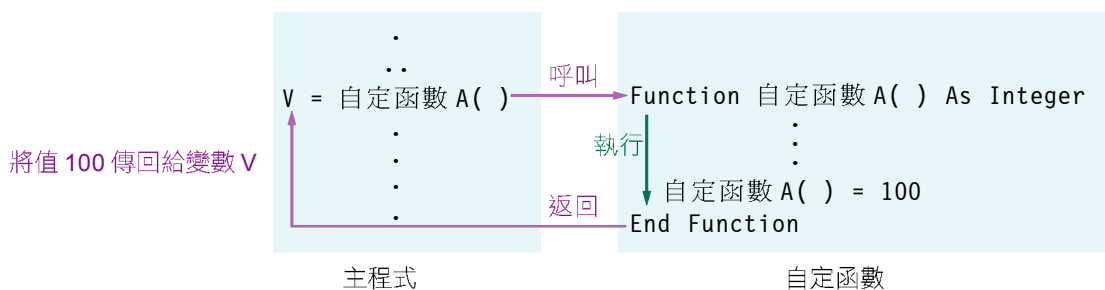
12 輸入程式碼，顯示系統的日期與時間

自定函數

腦力激盪

你能分辨副程式及自定函數的使用時機嗎？

自定函數的語法與副程式非常類似，它是由 **Function** 與 **End Function** 之間的程式敘述所組成。每個自定函數也都是一個專屬的名稱，當程式中的其它區塊需要使用這個函數功能時，便可以呼叫的方式來執行函數；函數在執行結束後，會回傳一個傳回值給呼叫它的程式敘述（如圖 12-60 所示）。



▲ 圖12-60 主程式呼叫自定函數的示意圖

自定函數的撰寫及呼叫之語法說明如下：

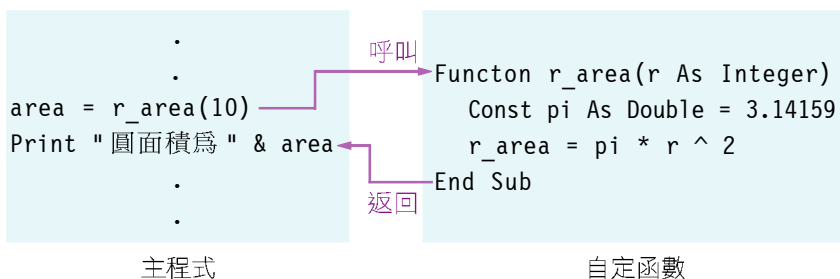
語法 1： **Function** 自定函數名稱(參數 1 **As** 資料型別, 參數 2 **As** 資料型別, …) **As** 資料型別
 程式敘述區塊
 自定函數名稱 = 傳回值
End Function

語法 2： 變數名稱 = 自定函數名稱(參數 1, 參數 2, …)

說明：

1. 語法 1 為設計自定函數使用的語法；語法 2 為呼叫自定函數使用的語法。
2. 在 **Function** 敘述中可加入要傳送給自定函數處理的**參數**（parameters），並設定參數的資料型別。
3. **Function** 敘述的最後可以宣告自定函數的資料型別，即傳回值的資料型別。
4. 呼叫自定函數時可以同時將參數值傳給自定函數，以便自定函數處理。

圖 12-61 為在主程式中呼叫自定函數 `r_area`，並將參數值 10 傳給該自定函數，以計算圓面積的程式實例。



▲ 圖12-61 自定函數的實例

自定函數的設計實例

步驟 1：定義問題

依據使用者輸入的西元年份，判斷該年份所屬的生肖，並顯示生肖的個性特質。

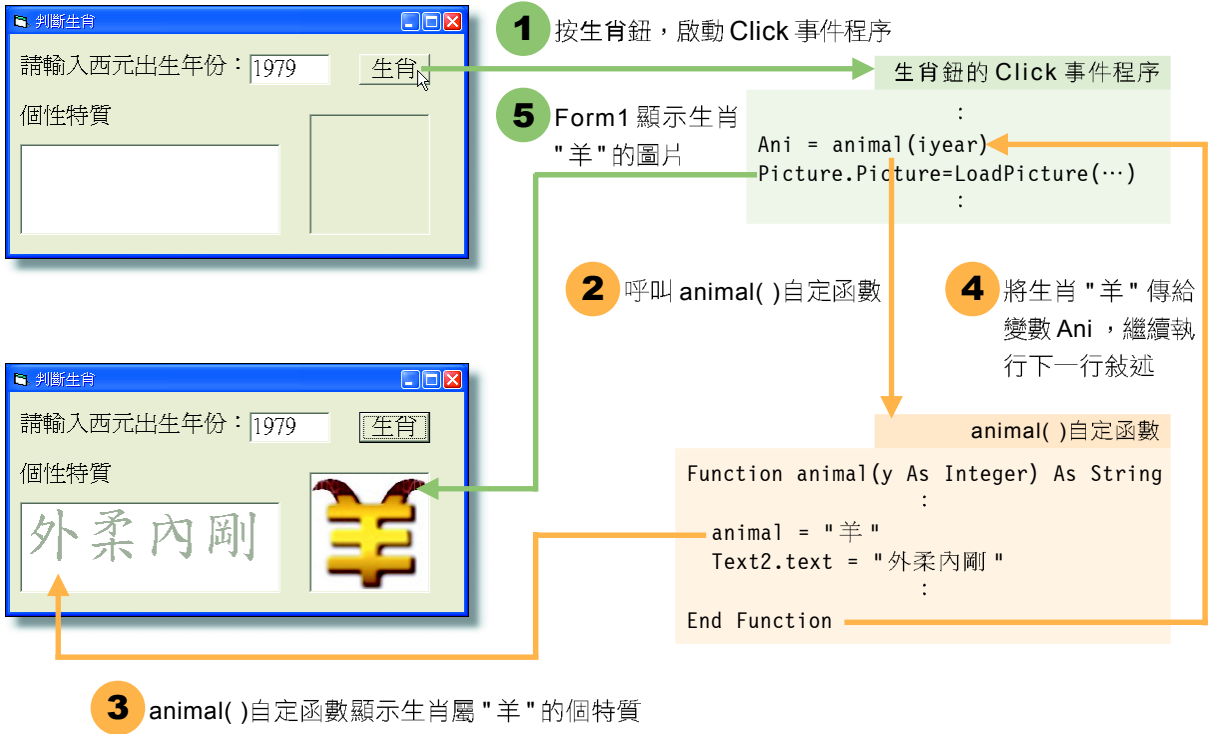
步驟 2：擬定解題計畫

- (1) 當使用者按下**生肖**鈕時，會先判斷使用者是否已輸入西元年份，及所輸入的西元年份是否為正整數，然後再呼叫自定函數 `animal()` 判斷生肖別。
- (2) `animal()` 自定函數會依照使用者所輸入的西元年份判斷該年份所屬的生肖，並顯示該生肖的個性特質。
- (3) 依照自定函數 `animal()` 傳回的生肖別；在表單上載入代表該生肖的圖片。
- (4) `animal()` 自定函數判斷生肖及顯示個性特質的執行過程示意圖（圖 12-62）及流程圖（圖 12-63）如下：

腦力激盪

下列程式執行後，表單上會顯示什麼結果？

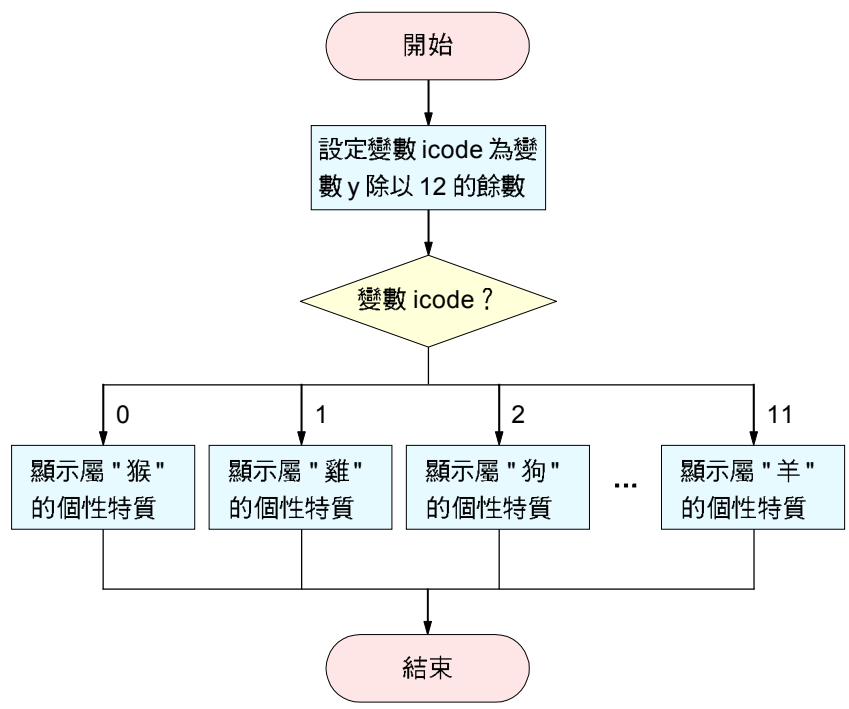
1. Private Sub Form_Activate()
 - x = 10
 - Print test(x)
 End Sub
 Function test(i)
 test = i
 i = i + 10
 End Function
2. Private Sub Form_Activate()
 - Print Sun(2, 10)
 End Sub
 Function Sun(X%, Y%)
 Sun = X ^ Y
 End Function



▲ 圖 12-62 `animal()` 自定函數判斷生肖及顯示個性特質的執行過程示意圖

小百科

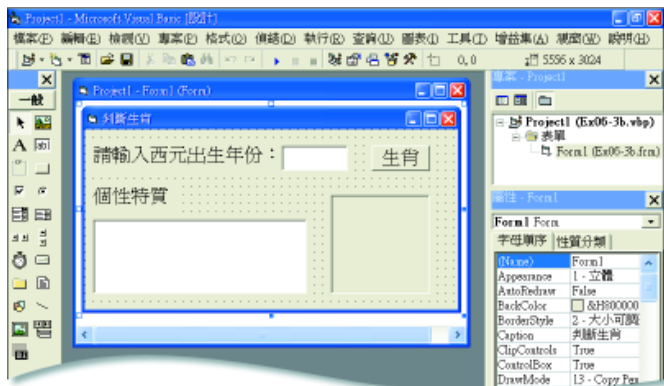
由經驗得知，西元出生年份除以 12 時，若餘數為 0，表示該年份應屬 "猴"；餘數為 1，表示該年份應屬 "雞"；餘數為 2，表示該年份應屬 "狗"，依此類推。



▲ 圖 12-63 `animal()` 自定函數判斷生肖及顯示個性特質的流程圖

步驟 3：撰寫程式

請同學開啓已佈妥控制物件的範例檔案 "Ex06-3b.vbp"（如圖 12-64 所示），並依照操作實例 9 的說明，撰寫「判斷生肖」的程式。



▲ 圖12-64 範例檔案Ex06-3b.vbp

操作實例 9

「判斷生肖」程式設計

撰寫一個「判斷生肖」的程式，然後測試程式的執行結果。

此條件式用來判斷使用者輸入的西元年份 (iyear) 是否為正整數；當轉換成數值型別的 Text 屬性值不等於整數型別的 Text 屬性值時，即表示使用者輸入的西元年份 (iyear) 不為整數

```

Project1 - Form1 (程式碼)
Command1 Click
Private Sub Command1_Click()
10 Dim iyear As Integer, Ani As String
20 iyear = Val(Text1.Text)
30 If Text1.Text = "" Then
40     MsgBox "請先輸入出生年份!", 48, "錯誤訊息"
50 Else
60     If iyear <= 0 Or Val(Text1.Text) <> Int(Val(Text1.Text)) Then
70         MsgBox "您所輸入的年份有誤, 請重新輸入", 48, "錯誤訊息"
80     Else
90         Ani = animal(iyear)
100        Picture1.Picture = LoadPicture(App.Path & "\\" & Ani & ".jpg")
110     End If
120 End If
End Sub

```

Step 1 在 Command1 的 Click 事件程序中撰寫判斷使用者輸入的年份是否合理並呼叫自定函數 animal()

1.1 輸入程式碼 10，宣告變數

1.2 輸入程式碼 20，將使用者輸入的西元年份傳給變數 iyear

1.3 輸入程式碼 30~70，判斷使用者輸入的西元年份是否合理？

1.4 輸入程式碼 80~120，呼叫自定函數 animal() 以取得生肖別，並載入該生肖的圖檔

Step 2 撰寫自定函數animal()

2.1 輸入程式碼，設定自定函數 `animal()` 的起始敘述

2.2 輸入程式碼 10~20 設定變數 `icode`，並設定變數 `icode` 為變數 `y` 除以 12 的餘數

2.3 輸入程式碼 30~40，利用變數 `icode` 判斷所屬生肖，顯示該生肖的個性特質，並將生肖別傳回

！ 小提示

當我們輸入自定函數的起始敘述後，只要按下 **Enter** 鍵，程式碼視窗即會自動顯示自定函數的結尾敘述（即 "End Function"）。

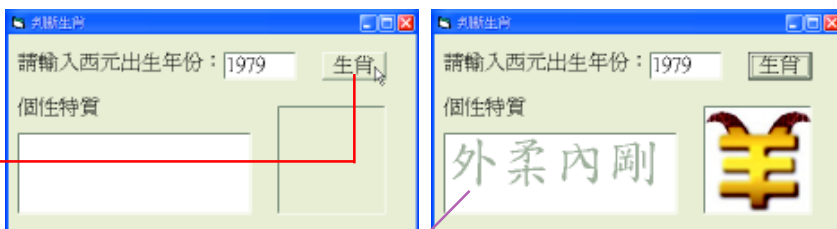
```

Function animal(y As Integer) As String
10  Dim icode As Integer
20  icode = y Mod 12
30  Select Case icode
    Case 0
        animal = "猴"
        Text2.Text = "才智高超"
    Case 1
        animal = "雞"
        Text2.Text = "好勝專注"
    Case 2
        animal = "狗"
        Text2.Text = "純樸正直"
    Case 3
        animal = "豬"
        Text2.Text = "真誠正直"
    Case 4
        animal = "鼠"
        Text2.Text = "機智多謀"
    Case 5
        animal = "牛"
        Text2.Text = "腳踏實地"
    Case 6
        animal = "虎"
        Text2.Text = "做事積極"
    Case 7
        animal = "兔"
        Text2.Text = "心思細密"
    Case 8
        animal = "龍"
        Text2.Text = "精力旺盛"
    Case 9
        animal = "蛇"
        Text2.Text = "冷靜沈著"
    Case 10
        animal = "馬"
        Text2.Text = "浪漫熱情"
    Case 11
        animal = "羊"
        Text2.Text = "外柔內剛"
40  End Select
End Function
  
```

Step 3 測試程式的執行結果

3.1 按開始鈕 ▶ 執行程式

3.2 輸入 "1979" 並按生肖鈕



顯示該年份所屬之生肖及該生肖的個性特質

馬上練習

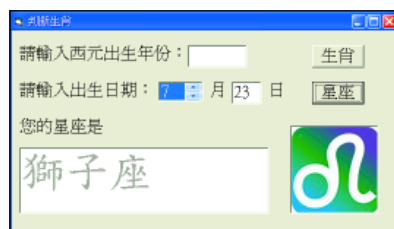
一、實作題

1. 請延續操作實例9，開啓範例檔案 "Ex06-e.vbp"，撰寫一個具有判斷星座功能的副程式，並將檔案儲存為 "Sp06-e.vbp"。

提示：(1) 利用 If-Then-Else 敘述來判斷使用者輸入之日期是否合理（大月 1、3、5、7、8、10、12 月有 31 天，2 月有 28 天，其餘月份有 30 天）。

- (2) 利用 Select Case 判斷敘述來判斷使用者輸入之月份，再利用 If-Then-Else 判斷敘述來判斷使用者輸入之日期，以辨識出所屬星座。下表為各星座所對應的出生日期範圍：

星座	出生日期範圍	星座	出生日期範圍
牡羊座	03/21~04/20	天秤座	09/23~10/22
金牛座	04/21~05/20	天蠍座	10/23~11/21
雙子座	05/21~06/21	射手座	11/22~12/21
巨蟹座	06/22~07/22	魔羯座	12/22~01/19
獅子座	07/23~08/22	水瓶座	01/20~02/19
處女座	08/23~09/22	雙魚座	02/20~03/20



12-3 節練習

一、選擇題

- @ () 1. 下列有關結構化程式設計之敘述，何者錯誤？ (A)儘量不採用GOTO敘述 (B)選擇結構是程式基本結構之一 (C)程式只有一個入口但可以有多個出口 (D)由上而下的程式設計。
- @ () 2. 假設樂透彩券的中獎號碼是10~42，在Visual Basic 語言中，下列何者可以產生一個 10~42 之間的整數亂數？ (A)Int(Rnd * 10)+ 33 (B)Int(Rnd * 32)+ 10 (C)Int(Rnd * 33)+ 10 (D)Int(Rnd * 41)+ 2。
- @ () 3. 在Visual Basic中，Print Mid("Hello", 3, 2)若與Print Left(Right("Hello", ___), ___)有相等作用，則空白處應分別填入 (A)2,2 (B)2,3 (C)3,2 (D)3,3。
- @ () 4. 在Visual Basic中，若要刪除A\$字串左邊的第1個字元，則其敘述可撰寫成 (A) A\$ = Left(A\$, 1) (B)A\$ = Right(A\$, 1) (C)A\$ = Mid(A\$, 1, 1) (D)A\$ = Right(A\$, Len(A\$) - 1)。
- @ () 5. 執行下列Visual Basic程式，則呼叫Test(1, -4, 3)會得到的結果為何？ (A)0 (B) 1 (C)2 (D)3。

```
Function Test(a, b, c) As Single
```

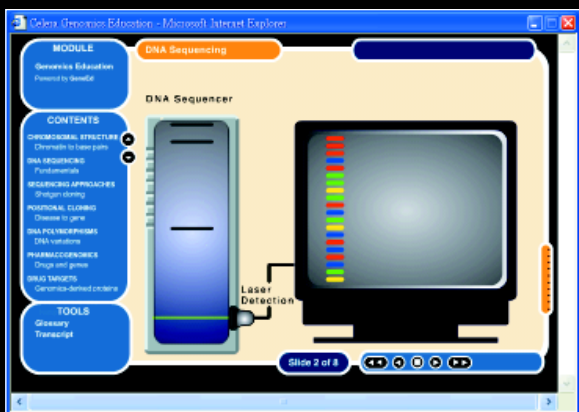
```
    Test = ((-b) + (b ^ 2 - 4 * a * c) ^ 0.5) / (2 * a)
```

```
End Function
```

科技脈動

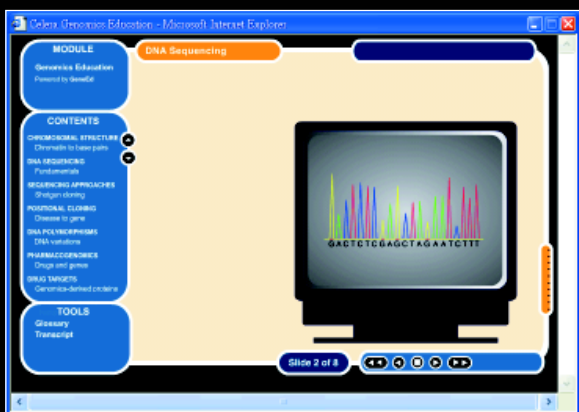
程式設計在科學上的應用

電腦軟體發展的目的是為了協助人類解決或處理特定的問題。由於軟體開發的技術及硬體設備的功能不斷突破，電腦軟體的應用範圍除了應用在一般的事務處理與決策制定等方面之外，也早已深入科學研究的領域。



A-1

(基因工程教學網頁)

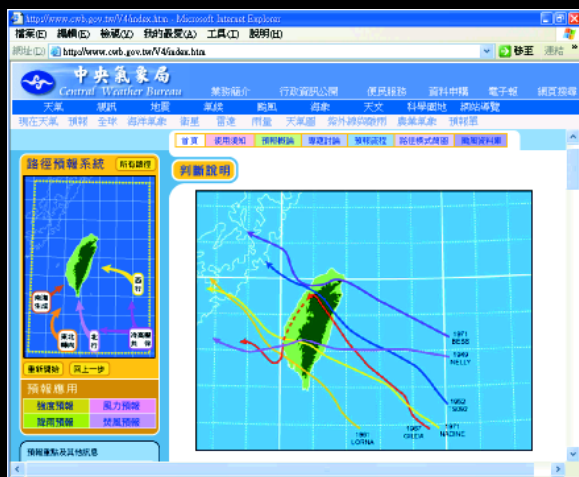


A-2

(基因工程教學網頁)

A 基因工程 利用超級電腦及大型軟體程式來整理、排列、及繪製人類的基因圖譜，以作為遺傳工程、疾病防治、基因改造等醫療研究的參考。

B 颱風預報系統 利用軟體系統將新形成的颱風與過去的颱風進行各種資料的比對，可判斷新颱風的可能路徑、強弱、及特性，以掌握其發展動向。



B

(介紹颱風預報系統的網頁)

- C 掩體穿透彈** 利用全球衛星定位系統及紅外線搜尋系統來找出攻擊目標，再使用功能強大的程式軟體控制鑽地的工作，以完成攻擊的任務。
- D 無人戰機** 只要將飛行座標輸入程式中，無人戰機就能在程式的控制下自行飛至目的地，並接收指令進行連繫、偵測、及攻擊的工作。
- E 火星的「航海家谷」影像** 將雷射光束照射在火星上，再根據光束反射時間之長短，利用電腦系統轉換為地勢的高低，繪製成火星地形圖。
- F 虛擬考古** 利用電腦斷層掃描器掃描深埋在地下的骨骸，再將所得的立體影像與電腦中龐大的資料庫進行比對，可判斷骨骸的地理族群及生存年代。



(介紹鑽體穿透彈的網頁)



(courtesy of NASA)

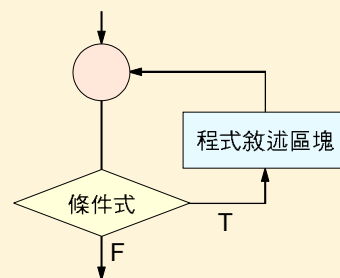
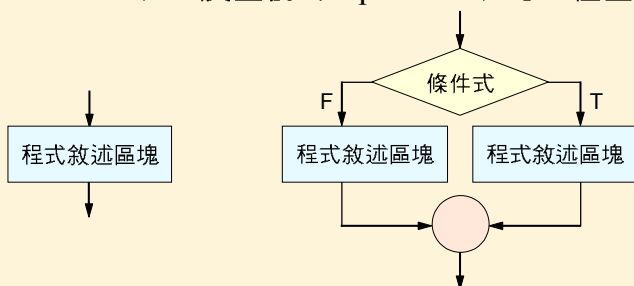
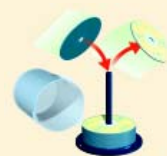


(介紹虛擬考古的網頁)



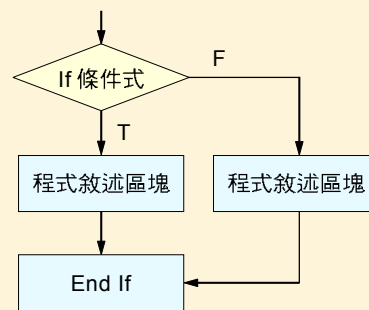
(介紹拍攝火星影像的網頁)

- 資料結構 (data structure) 是用來組織及管理資料的結構設計；可用來定義資料的放置方法及存取規則。
- 陣列 (array) 是由一群具有順序關係之相關資料元素所組成的資料結構。
- 堆疊 (stack) 是一種具有後進先出 (Last In First Out, LIFO) 特性的資料結構。
- 佇列 (queue) 是一種具有先進先出 (First In First Out, FIFO) 特性的資料結構。
- 一個程式大多是由循序 (sequence)、條件 (condition)、及重覆 (repetition) 等 3 種基本結構所組成。



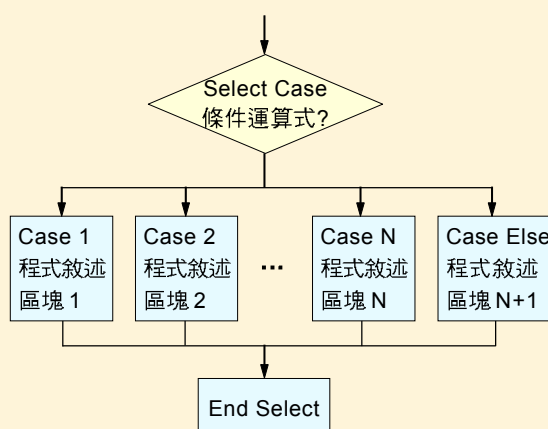
- **If-Then-Else** 是一種條件判斷的敘述，其語法如下：

語法：If 條件式 Then
 條件式為真 (True) 程式敘述區塊
 Else
 條件式為假 (False) 程式敘述區塊
 End If



- **Select Case** 敘述是專門用來處理多重選擇的條件判斷敘述，其語法如下：

語法：Select Case 資料或運算式
 Case 條件式 1
 程式敘述區塊 1
 Case 條件式 2
 程式敘述區塊 2
 ⋮
 Case 條件式 N
 程式敘述區塊 N
 Case Else
 程式敘述區塊 N+1
 End Select



- **For-Next迴圈**敘述可用來處理程式中需重覆執行，或已確定執行次數的某段程式敘述，其語法如下：

語法：For 控制變數 = 起始值 To 終止值 Step 增值（減值）
 程式敘述區塊
 Next 控制變數

- **While-Wend迴圈**敘述可用來處理程式中不確定執行次數的某段程式敘述，其語法如下：

語法：While 條件式
 程式敘述區塊
 Wend

- **結構化程式設計**主要是遵循由上到下的程式設計、模組化的程式設計、及使用三種程式基本結構等三個原則來進行程式設計的一種技術。

- **模組（module）**是指一段具有某種特定功能、大小適中、容易閱讀、修改、及維護的程式。使用**副程式（subprogram）**及**函數（function）**可將程式分割成多個功能單純的模組，以利程式的撰寫、測試、及維護。

- **副程式**是一段獨立的程式區塊；當程式中的其它區塊需要用到副程式的功能時，可以使用呼叫（call）的方式來執行副程式。其語法如下：

定義副程式：Sub 副程式名稱(參數 1 As 資料型別, 參數 2 As 資料型別, …)
 程式敘述區塊
 End Sub

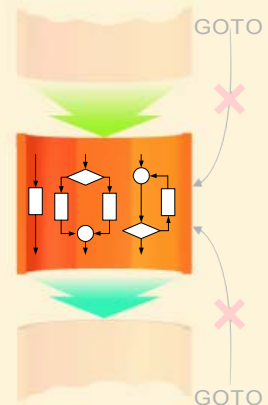
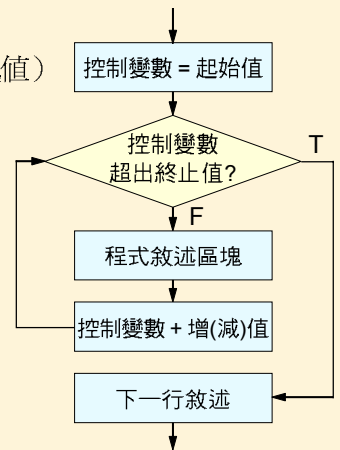
呼叫副程式：Call 副程式名稱(參數 1, 參數 2, …)

- 函數分為**內建函數（built-in function）**及**自定函數（user-defined function）**兩種，內建函數是程式語言預先設計好的程式，使用者可直接輸入函數名稱來使用其功能，而自定函數則是使用者依特定需求自行撰寫的函數。

- **自定函數**也是一段獨立的程式區塊；自定函數在執行結束後，會回傳一個傳回值給呼叫它的程式敘述。其語法如下：

定義自定函數：Function 自定函數名稱(參數 1 As 資料型別, …) As 資料型別
 程式敘述區塊
 自定函數名稱 = 傳回值
 End Function

呼叫自定函數：變數名稱 = 自定函數名稱(參數 1, 參數 2, …)



一、選擇題

- () 1. 有一個陣列宣告為 "Dim A(10) As Integer"，A(0) 的記憶體位址為 6，假設每一個陣列元素佔用 2bytes 記憶體，若此陣列是連續存放在記憶體中，則 A(5) 的記憶體位置應為下列何者？ (A) 6 (B) 12 (C) 14 (D) 16。
- () 2. 電影院排隊買票，排在最前方的人會先獲得服務，後到者須排在隊伍的最後面等待，這是下列哪一種資料結構的應用？ (A) 陣列 (B) 堆疊 (C) 佇列 (D) 字串。
- () 3. 程式設計包含以下步驟：①測試與除錯②製作文件③擬定解題計畫④撰寫程式⑤定義問題，其正確順序為 (A) ①②③④⑤ (B) ⑤③④①② (C) ⑤④③②① (D) ①③②⑤④。
- @ () 4. If-Then-Else 為下列何種結構？ (A) 循序結構 (B) 選擇結構 (C) 重覆結構 (D) 以上皆非。
- @ () 5. 下列哪一個 Visual Basic 敘述最適合用在多個選擇的決策中？ (A) If-Then-Else (B) For-Next (C) Select Case (D) Do-Loop-Until。
- @ () 6. 執行下面 Visual Basic 程式片段後，變數 C 的值為何？ (A) C = 1 (B) C = 2 (C) C = 3 (D) C = 4。
- ```

A = 1: B = 2: C = 3
If B > 1 Then
 If A < 1 Then
 C = 2
 Else
 C = 4
 End If
End If

```
- ( ) 7. 下列何種程式語法可以無條件跳到程序中的某標記或行號？ (A) Select Case (B) GoTo (C) For-Next (D) While-Wend。
- @ ( ) 8. 執行下面 Visual Basic 程式片段，程式輸出的結果為何？ (A) 0 (B) -1 (C) -2 (D) -3。
- ```

K = 0
For I = 1 To 5
    For J = I + 2 To 4
        K = K + 1
    Next J
    K = K - 1
Next I
Print K

```

- () 9. 欲列印整數 10 之所有因數，則空白處應填入什麼？ (A) $> N$ (B) $\geq N$ (C) $< N$ (D) $\leq N$ 。

```
N = 10 : I = 0
While I _____
    I = I + 1
    If N Mod I = 0 Then Print I
Wend
```

- () 10. 有關 Visual Basic 內建函數的傳回值，下列何者正確？ (A)`Left("明天會更好", 2) = 更好` (B)`Fix(-5.8) = -5` (C)`Abs(5.8) = 5` (D)`Sqr(9) = 81`。

- @ () 11. 有關副程式 (Subprogram) 的敘述，下列何者不正確？ (A) 可使主程式易於維護 (B) 副程式之中可再去執行另一個副程式 (C) 每一個主程式都必須要有副程式 (D) 若程式中有某部份需經常用到者可將其寫成副程式。

- @ () 12. 執行下列 Visual Basic 程式，則呼叫 `FNF(3, 4)` 會得到的結果為何？ (A) 625 (B) 5 (C) 25 (D) 7。

```
Function FNF(A, B)
    FNF = SQR(A ^ 2 + B ^ 2)
End Function
```

二、多元練習題

1. 假設 S 為宣告可儲存五個資料元素的堆疊，程式 I 具有自堆疊 S 頂端推入資料 (push) 的功能，程式 O 具有自堆疊頂端彈出資料 (pop) 的功能。如果我們要將 DEER (鹿) 這個英文字重新排列成 REED (蘆葦)，則利用程式 I 及程式 O 執行堆疊 S 之資料推入與彈出，其運算順序為何？

2. 請開啟已佈建控制物件的範圍檔案 "Ex06-f.vbp"，利用一維陣列及迴圈敘述設計一個具有排序功能的程式，最後將檔案儲存為 "Sp06-f.vbp"。

提示：(1) 排序的演算法可運用第 11 章所介紹的選擇排序法或氣泡排序法。

- (2) 利用文字方塊控制物件陣列來輸入所要排序的數字，並顯示排序後的數字。

