

程式語言(programming language)

使用Visual Basic 劉和師

Part 3

陣列(Array)

- ▶ 用途：
 - ▶ Array係用來儲存一組具有相同資料型態的資料
- ▶ 目的：
 - ▶ 減少變數的個數,使程式的編寫更容易
 - ▶ 陣列可配合迴圈來處理大量資料

陣列(Array)

- ▶ 為何要使用陣列?
 - ▶ 若要將一組有相關性意義的資料(如全年中各月份的薪資)存放在記憶體中,若每一份資料宣告一個變數,如:

```
Dim salary1 As Double    '一月份薪水
Dim salary2 As Double    '二月份薪水
                        :
Dim salary12 As Double    '十二月份薪水
```

- ▶ 如此宣告使得程式的編寫變得冗長而不具效率
- ▶ 故應使用陣列來改進此種缺失

一維陣列

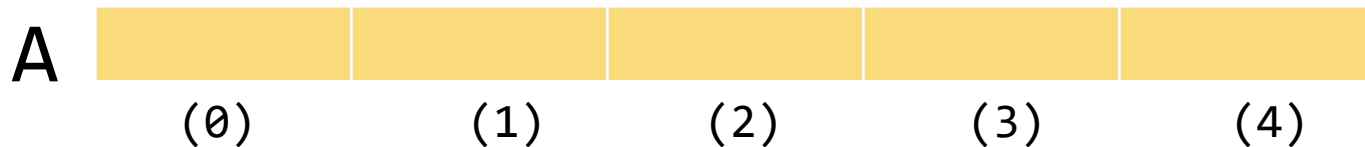
- ▶ 宣告：任何陣列在使用前必須先宣告,格式為

Dim 陣列名稱(最大註標值) **As** 資料型態

- ▶ Ex :

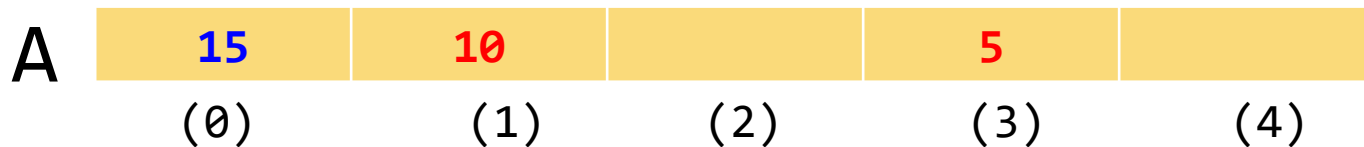
Dim A(4) As Integer

- ▶ 宣告了一個陣列，名稱為 **A**，共有5個元素，最大註標為4，皆存放整數，圖示如下



一維陣列

- ▶ 陣列是透過註標值(或稱索引值)來存取陣列的每個元素
- ▶ Ex :
 - ▶ $A(1) = 10$ '把10存入A(1)位置
 - ▶ $A(3) = 5$ '把5存入A(3)位置
 - ▶ $A(0) = A(1) + A(3)$ '把A(1)與A(3)的內容相加後存入A(0)



一維陣列

- ▶ 也可以在陣列宣告時即指定初值：

```
Dim A( ) As Integer = {10, 2, 35, 41}
```

↑
空著不寫

↑ ↑ ↑ ↑
A(0) A(1) A(2) A(3)

- ▶ 系統會自動指定陣列的大小為A(3)
- ▶ 若要存取A(4)則會發生錯誤

一維陣列

- ▶ 要記得註標總是從0開始
- ▶ 所以若宣告

Dim A(5) As Integer

- ▶ 其實有6個元素

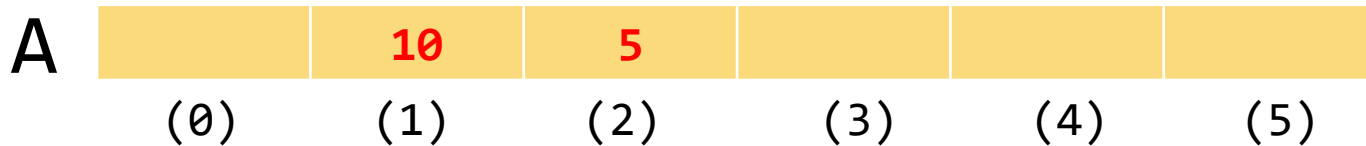
A(0), A(1), A(2), A(3), A(4), A(5)

- ▶ 存取大於註標值時會發生錯誤，例如A(6)是錯誤的

一維陣列

- ▶ 註標值可以是一個變數或運算式，只要在合法的範圍內就可以了
- ▶ Ex :

```
Dim A(5), X As Integer  
X = 1  
A(X) = 10  
A(X+1) = 5
```



一維陣列

- ▶ 所以陣列很容易交給迴圈去處理
- ▶ Ex :

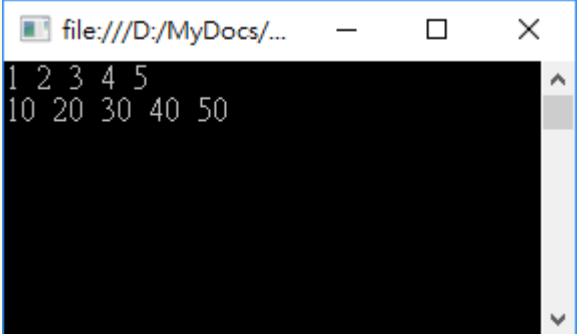
```
For I = 1 to 10  
    A(I) = A(I) + 1  
Next
```

- ▶ 將陣列內所有元素的值+1(假定10個元素)

一維陣列

- ▶ 將陣列內容都乘以10：

```
Dim A() As Integer = {1, 2, 3, 4, 5}
For i = 0 To 4    '印出原陣列
    Console.Write(A(i) & " ")
Next
For i = 0 To 4    '每個元素*10
    A(i) = A(i) * 10
Next
Console.WriteLine()
For i = 0 To 4    '印出改變後的陣列內容
    Console.Write(A(i) & " ")
Next
Console.Read()
```



```
file:///D:/MyDocs/...
1 2 3 4 5
10 20 30 40 50
```

一維陣列

- ▶ 找出陣列中最大值：

```
Dim A() As Integer = {13, 25, 33, 14, 5}
Dim max As Integer = 0
For i = 0 To 4
    If A(i) > max Then max = A(i)
Next
Console.WriteLine(max)
Console.Read()
```

For Each敘述

- ▶ For Each是專門為了陣列元素而設計的敘述，無需使用者判斷陣列的邊界或索引值，增加了效率和方便性。
- ▶ 語法：

```
For Each 變數名稱 In 陣列名稱  
    :  
    敘述  
    :  
Next
```

- ▶ 迴圈會依序將陣列元素提至變數之內以供處理

For Each敘述

- ▶ 改寫找出陣列中最大值：

```
Dim A() As Integer = {13, 25, 33, 14, 5}
Dim max, num As Integer
max = 0
For Each num In A
    If num > max Then max = num
Next
Console.WriteLine(max)
Console.Read()
```



一維陣列練習

- ▶ 一、請實作Page8-5的練習
- ▶ 提示：
 - ▶ 函數UBound()是取得陣列的最大索引值，例如A(5)，則
 - ▶ UBound(A)會得到5。
 - ▶ 但記得總元素是6個喔，因為包含0那個
 - ▶ 還有一個函數LBound()，取得陣列的最小索引值，不過通常是0，所以較少用到
- ▶ 二、將本題改寫成由使用者輸入成績

二維陣列

- ▶ 一個棋盤，如果我們要指定其中的每一格，通常用座標，例如：

	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

- ▶ 這就是一個二維陣列

二維陣列

- ▶ 二維陣列宣告方式：

Dim 變數名稱(註標一, 註標二) **As** 資料型態

- ▶ Ex：

Dim A(2, 2) As Integer

A(0,0)	A(0,1)	A(0,2)
A(1,0)	A(1,1)	A(1,2)
A(2,0)	A(2,1)	A(2,2)

二維陣列

- ▶ 二維陣列初值設定：

```
Dim A(, ) As Integer = { {N1,N2} , {N3,N4} }
```

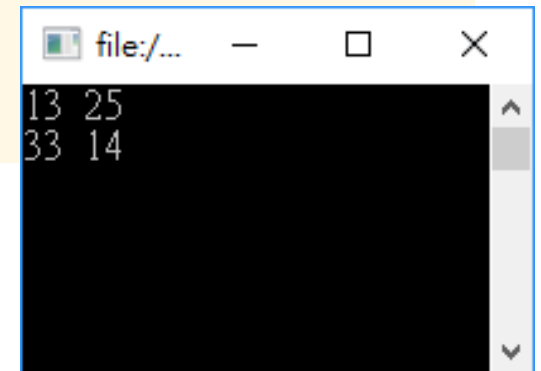
- ▶ 陣列A內容：

$A(0,0)$ N1	$A(0,1)$ N2
$A(1,0)$ N3	$A(1,1)$ N4

二維陣列

- 指定二維陣列初值並列印出來：

```
Dim A(,) As Integer = {{13, 25}, {33, 14}}
For i = 0 To 1
    For j = 0 To 1
        Console.Write(A(i, j) & " ")
    Next
    Console.WriteLine()
Next
Console.Read()
```



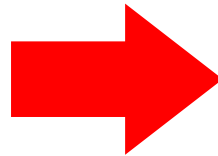
A screenshot of a Windows console window. The title bar shows 'file:/...' and standard window controls. The console output displays two lines of text: '13 25' on the first line and '33 14' on the second line. The text is white on a black background.

二維陣列練習

▶ 練習：陣列轉置(一)

1	2	3
4	5	6
7	8	9

A

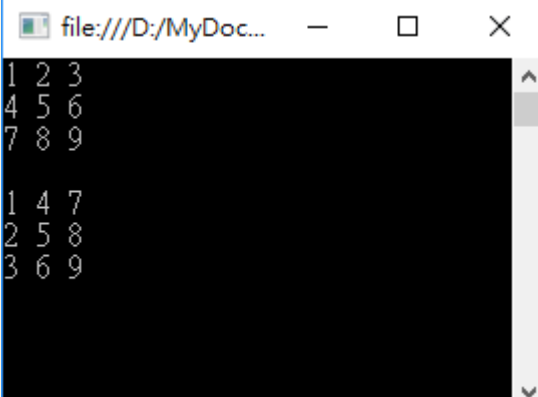


1	4	7
2	5	8
3	6	9

B

二維陣列練習

```
Dim A(,) As Integer = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}
Dim B(2, 2) As Integer
For i = 0 To 2    '印出A陣列
    For j = 0 To 2
        Console.Write(A(i, j) & " ")
    Next
    Console.WriteLine()
Next
Console.WriteLine()
For i = 0 To 2    '轉置存入B陣列
    For j = 0 To 2
        B(j, i) = A(i, j)
    Next
Next
For i = 0 To 2    '印出B陣列
    For j = 0 To 2
        Console.Write(B(i, j) & " ")
    Next
    Console.WriteLine()
Next
Console.Read()
```



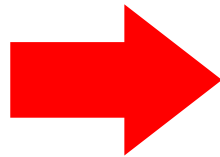
```
file:///D:/MyDoc...
1 2 3
4 5 6
7 8 9
1 4 7
2 5 8
3 6 9
```

二維陣列練習

▶ 練習：陣列轉置(二)

1			
1	2		
1	2	3	
1	2	3	4

A



1	1	1	1
	2	2	2
		3	3
			4

A

二維陣列練習

```
Dim A(4, 4), temp As Integer
For i = 1 To 4      '填入陣列內容
    For j = 1 To i
        A(i, j) = j
    Next
Next
For i = 1 To 4      '印出陣列
    For j = 1 To 4
        If A(i, j) <> 0 Then
            Console.Write(A(i, j) & " ")
        Else
            Console.Write(" ")
        End If
    Next
    Console.WriteLine()
Next
```

有兩處「印出陣列」的程式碼是一模一樣的，後面會介紹使用副程式來縮減程式碼

二維陣列練習

```
For i = 1 To 4 '陣列轉置
    For j = 1 To i
        temp = A(i, j)
        A(i, j) = A(j, i)
        A(j, i) = temp
    Next
Next
```

```
Next
```

```
Console.WriteLine()
```

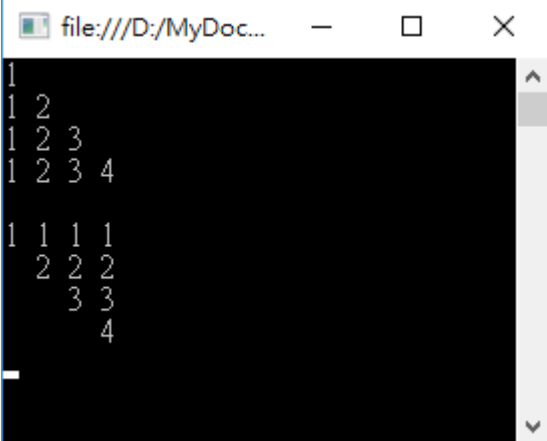
```
For i = 1 To 4 '印出陣列
    For j = 1 To 4
        If A(i, j) <> 0 Then
            Console.Write(A(i, j) & " ")
        Else
            Console.Write(" ")
        End If
    Next
    Console.WriteLine()
```

```
Next
```

```
Console.WriteLine()
```

```
Next
```

```
Console.Read()
```



```
1
1 2
1 2 3
1 2 3 4

1 1 1 1
2 2 2
3 3
4
```

與前頁「印出陣列」的程式碼是一模一樣的，應使用副程式來縮減程式碼

多維陣列

- ▶ 通常二維陣列以上的皆稱為多維陣列
- ▶ 不過很少使用到三維以上的陣列，因為過多的維度概念上不容易思考及理解，容易造成程式錯誤
- ▶ 據說我們的宇宙有11個維度...



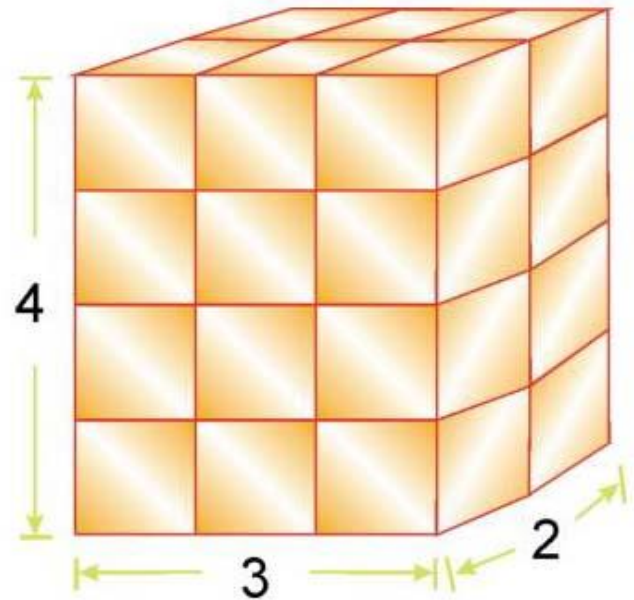
三維陣列

▶ 宣告：

Dim 陣列名稱(註標一,註標二,註標三) As 資料型態

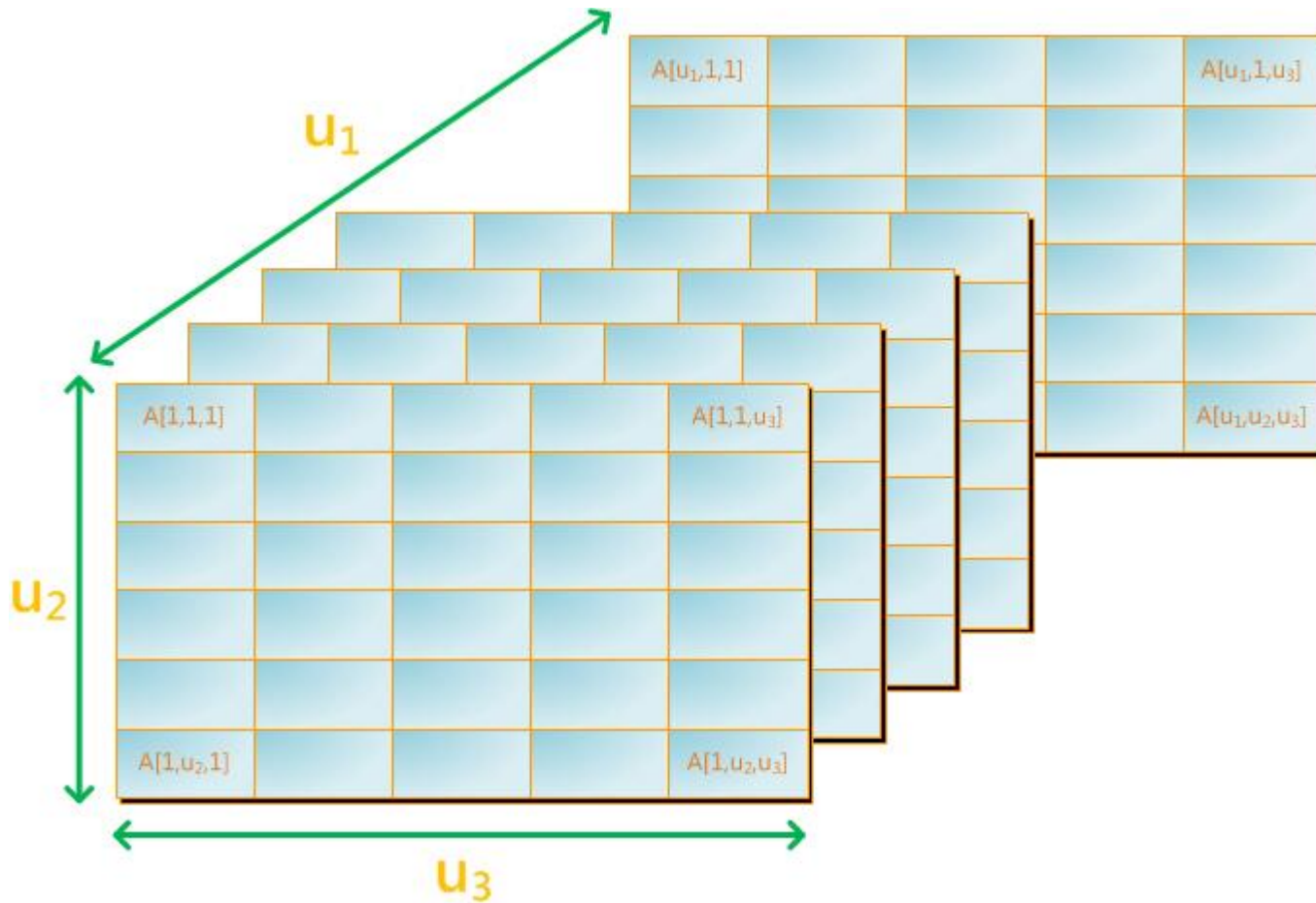
▶ 例如：

Dim A(4,3,2) As Integer



三維陣列

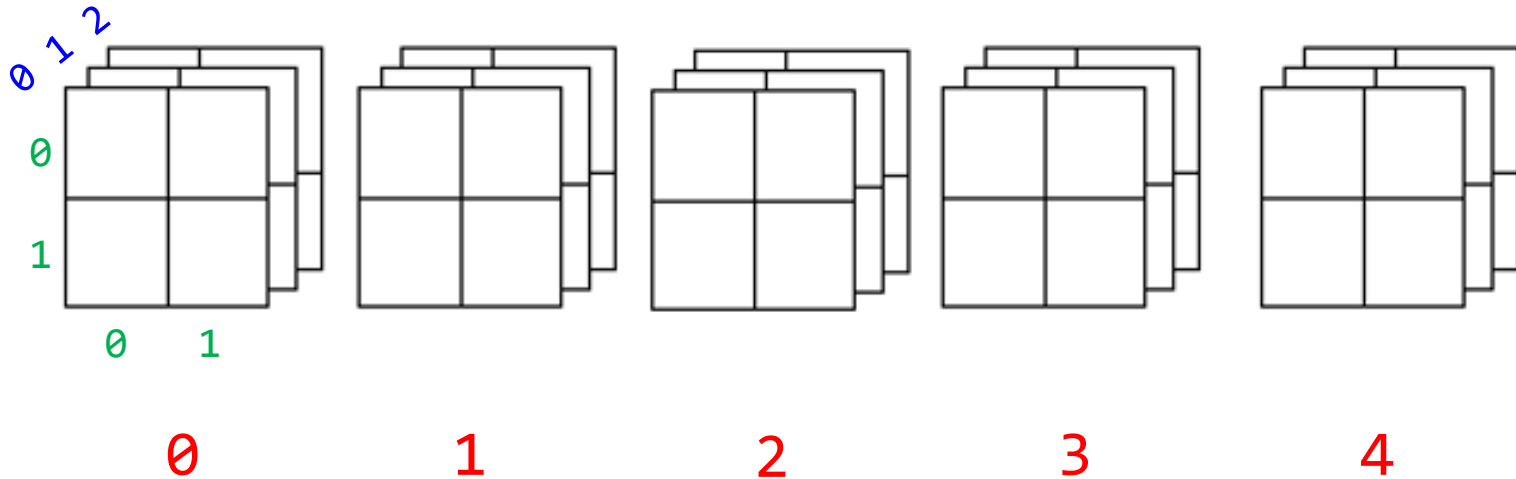
- ▶ 可想成是多個二維陣列



四維陣列

- ▶ 可想成是多個三維陣列
- ▶ 例如：

Dim A(**1**,**1**,**2**,**4**) As Integer



動態陣列

- ▶ 陣列在宣告時即已指定大小，無法於程式執行中改變
- ▶ 若資料數不定，宣告得太大會浪費記憶體空間，宣告得太小會造成執行錯誤，那怎麼辦呢？
- ▶ 當陣列大小可依需求增減者，稱為「動態陣列」
 - ▶ 方法一：使用變數決定
 - ▶ 方法二：使用ReDim敘述

動態陣列

► 方法一：使用變數決定陣列大小

```
Dim num As Integer
```

```
:
```

```
:
```

```
num = 10
```

```
:
```

```
:
```

```
Dim A(num) As Integer
```

‘先指定一個變數

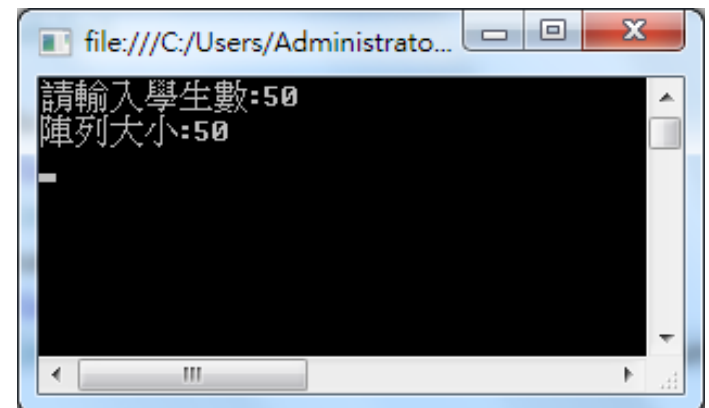
‘在程式中指定變數值

‘利用變數決定陣列大小

動態陣列

► 方法一：使用變數決定陣列大小範例

```
Dim num As Integer
Console.Write("請輸入學生數:")
num = CInt(Console.ReadLine())
Dim A(num) As Integer
Console.WriteLine("陣列大小:" & UBound(A))
Console.Read()
```



動態陣列

- ▶ 方法二：使用ReDim敘述
- ▶ 使用ReDim敘述即可重新定義陣列大小

▶ Ex：

```
Dim A(5) As Integer
```

‘宣告陣列大小為5

```
:
```

```
ReDim A(10)
```

‘重新宣告陣列大小為10

- ▶ 重新宣告後陣列內容將清空，全部重設為0

動態陣列

- ▶ 方法二：使用ReDim + Preserve敘述
- ▶ 使用Preserve指令即可保留原陣列內容
 - ▶ Ex：

```
Dim A(5) As Integer      ‘宣告陣列大小為5
:
ReDim Preserve A(10)     ‘重新宣告陣列大小為10
                        ‘並保留原陣列元素內容
```

- ▶ 若是增加，則多的部分初值為0
- ▶ 若是減少，則刪掉的部分其內容也丟失了

動態陣列

- ▶ ReDim僅能改變陣列大小，不能改變陣列維度

- ▶ Ex :

```
Dim A(5) As Integer
```

```
ReDim A(10)
```

```
ReDim A(5,5)
```

‘可以

‘錯誤，不能改變維度

- ▶ 對於多維陣列，ReDim僅能改變最後一個維度的大小，否則會有錯誤訊息

- ▶ Ex :

```
Dim A(1,2,3) As Integer
```

```
ReDim A(1,2,5)
```

```
ReDim A(1,4,3)
```

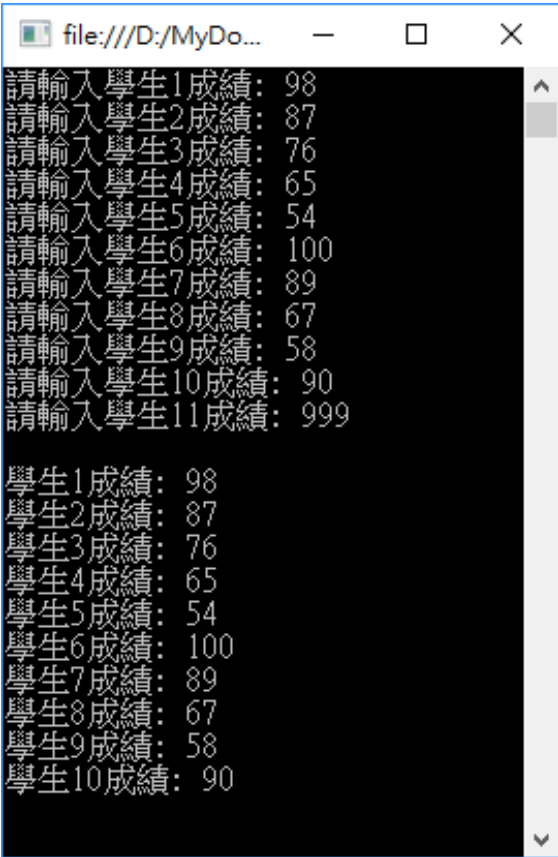
‘可以

‘錯誤，僅能改變最後一個維度

動態陣列練習

- ▶ 寫一程式，由鍵盤輸入學生成績，學生數目不確定，直到輸入999為止 (999不計入)。將剛才輸入的成績列印出來，執行結果如右圖

- ▶ 提示：
 - ▶ 1. 使用動態陣列
 - ▶ 2. 使用無盡迴圈或以999為離開迴圈的條件



```
file:///D:/MyDo...  
請輸入學生1成績: 98  
請輸入學生2成績: 87  
請輸入學生3成績: 76  
請輸入學生4成績: 65  
請輸入學生5成績: 54  
請輸入學生6成績: 100  
請輸入學生7成績: 89  
請輸入學生8成績: 67  
請輸入學生9成績: 58  
請輸入學生10成績: 90  
請輸入學生11成績: 999  
  
學生1成績: 98  
學生2成績: 87  
學生3成績: 76  
學生4成績: 65  
學生5成績: 54  
學生6成績: 100  
學生7成績: 89  
學生8成績: 67  
學生9成績: 58  
學生10成績: 90
```

動態陣列練習

```
Dim A(1) As Single
Dim num As Integer = 1
Do
    Console.Write("請輸入學生" & num & "成績: ")
    A(num) = CSng(Console.ReadLine())
    If A(num) = 999 Then
        Exit Do
    Else
        num += 1
        ReDim Preserve A(num)
    End If
Loop While 1
Console.WriteLine()
For i = 1 To UBound(A) - 1
    Console.WriteLine("學生" & i & "成績: " & A(i))
Next
Console.Read()
```

動態陣列練習

- ▶ 請將Page 8-11的練習做出來
- ▶ 將其修改成重新宣告陣列大小後仍保留原來的內容而不清空



陣列相關指令

- ▶ Erase：刪除陣列，釋放記憶體空間

```
Dim A(10) As Integer
      :
Erase A      ‘刪除陣列A
```

- ▶ Clone()：複製陣列

```
Dim A() As String = {"劉","和","師"}
Dim B(2) As String
      :
B = A.Clone()    ‘將A陣列的內容複製到B陣列
```

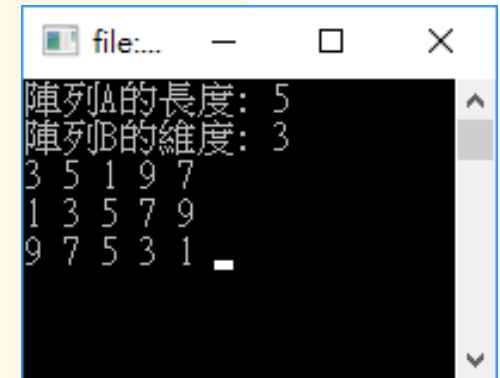
常用陣列屬性與方法

名 稱	說 明
Length	取得陣列長度
Rank	取得陣列維度
Clear	清除所有元素成預設值
Clone	複製整個陣列
Sort	將陣列進行排序
Reverse	反轉陣列中的元素
IndexOf	傳回陣列中第一個與指定值相符的元素索引
BinarySearch	對已排序的陣列進行二元搜尋，若找到則傳回索引值，否則就傳回負值

常用陣列屬性與方法

► Ex :

```
Dim A() As Integer = {3, 5, 1, 9, 7}
Dim B(2, 2, 2) As Integer
Console.WriteLine("陣列A的長度: " & A.Length)
Console.WriteLine("陣列B的維度: " & B.Rank)
For Each n In A      '印出原陣列
    Console.Write(n & " ")
Next
Console.WriteLine()
Array.Sort(A)
For Each n In A      '印出排序後陣列
    Console.Write(n & " ")
Next
Console.WriteLine()
Array.Reverse(A)
For Each n In A      '印出反轉後陣列
    Console.Write(n & " ")
Next
Console.Read()
```



```
file:...
陣列A的長度: 5
陣列B的維度: 3
3 5 1 9 7
1 3 5 7 9
9 7 5 3 1
```

排序和搜尋



3png.com

排序

- ▶ 排序(Sort)是將資料依某種規則重新安排其先後順序
- ▶ 最常見的是依大小或字母排序
- ▶ 常見排序方法
 - ▶ 一、氣泡排序法(Bubble Sort)
 - ▶ 二、選擇排序法(Selection Sort)
- ▶ 不同的排序方法效率不同，應用的地方也不同



排序

- ▶ 氣泡排序法(Bubble Sort)：

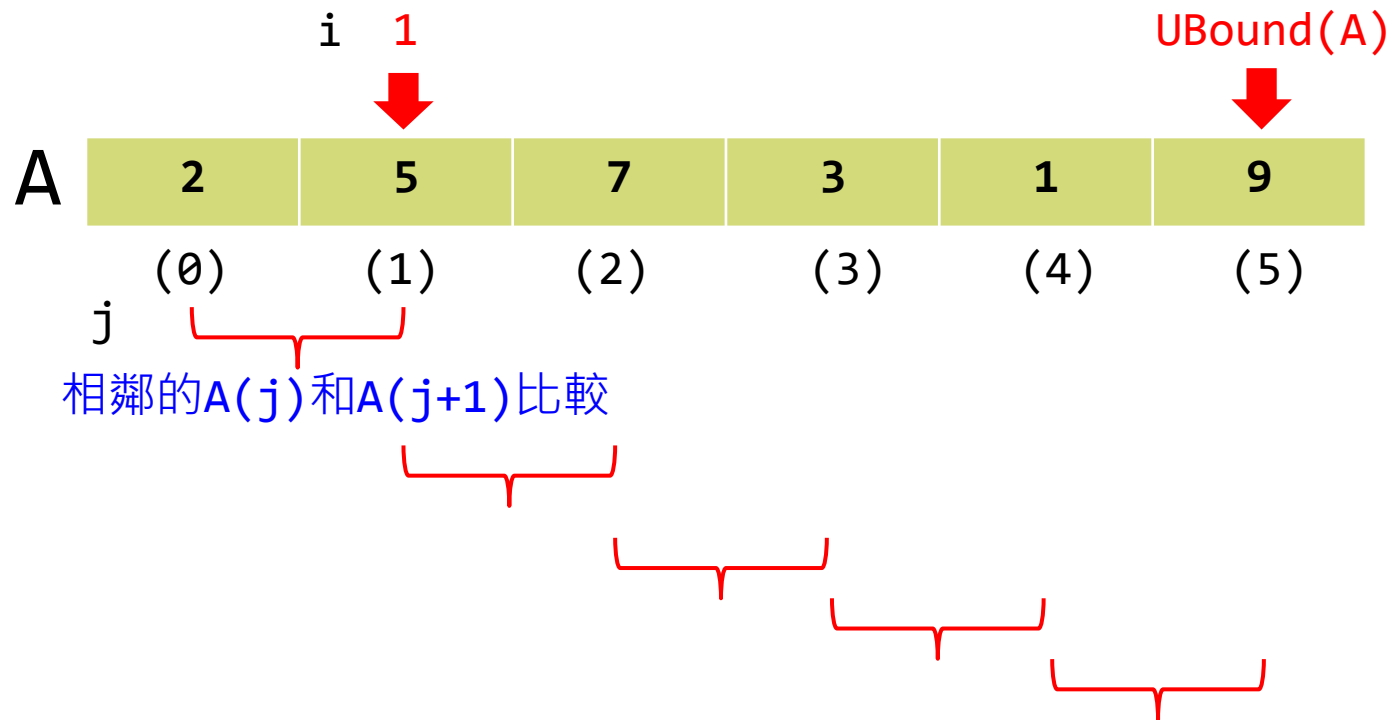
是一種簡單的排序演算法。它重複地走訪過要排序的數列，每次比較相鄰的兩個元素，如果他們的順序錯誤就把他們交換過來。

- ▶ 請參考：

<https://www.youtube.com/watch?v=JTPs8rZWBfl>

氣泡排序法(Bubble Sort)

- ▶ 假設原陣列內容為



- ▶ 每循環一次就確定一個元素位置

氣泡排序法(Bubble Sort)

► 排序迴圈：

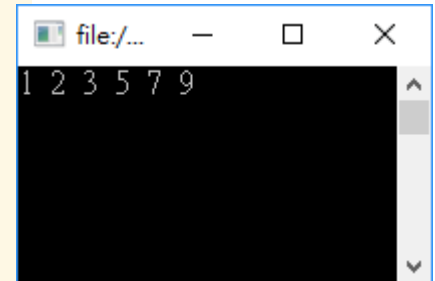
```
For i = 1 To UBound(A)
    For j = 0 To UBound(A) - i
        交換A(j)和A(j+1)
    Next
Next
```

氣泡排序法(Bubble Sort)

```
Dim A() As Integer = {2, 5, 7, 3, 1, 9}
Dim tmp As Integer
For i = 1 To UBound(A)
    For j = 0 To UBound(A) - i
        If A(j) > A(j + 1) Then
            tmp = A(j)
            A(j) = A(j + 1)
            A(j + 1) = tmp
        End If
    Next
Next
For i = 0 To UBound(A)
    Console.Write(A(i) & " ")
Next
Console.Read()
```

‘交換兩元素

‘印出陣列



氣泡排序練習

- ▶ 用亂數產生6個1~49之間的亂數，並由小至大排序後印出來
- ▶ 改變上述程式一個地方，使其成為由大至小排序

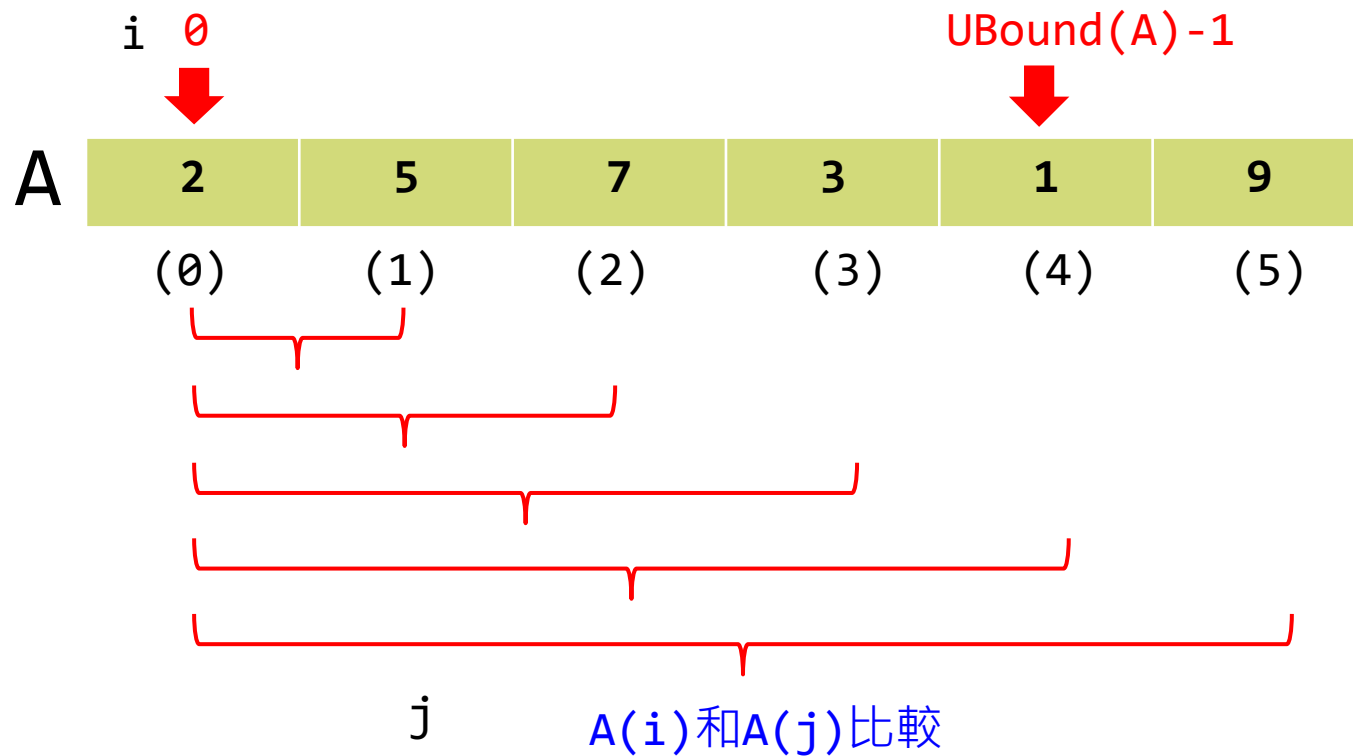
氣泡排序練習

```
Randomize()  
Dim A(5), tmp As Integer  
For i = 0 To 5      '產生6個亂數  
    A(i) = Int(Rnd() * 49) + 1  
Next  
For i = 1 To UBound(A)    '氣泡排序  
    For j = 0 To UBound(A) - i  
        If A(j) > A(j + 1) Then  
            tmp = A(j)  
            A(j) = A(j + 1)  
            A(j + 1) = tmp  
        End If  
    Next  
Next  
For i = 0 To UBound(A)    '印出陣列  
    Console.Write(A(i) & " ")  
Next  
Console.Read()
```

將>改成<即可
反向排序

選擇排序(Selection Sort)

- 與氣泡排序法類似，效率也相同，但比較兩元素的方式不同



選擇排序(Selection Sort)

► 排序迴圈：

```
For i = 0 To UBound(A)-1
    For j = i+1 To UBound(A)

        交換A(i)和A(j)

    Next
Next
```

選擇排序練習

- ▶ 用亂數產生6個1~49之間的亂數，並由小至大排序後印出來
- ▶ 提示：與氣泡排序大致相同，注意控制兩迴圈的參數及比較的方式

選擇排序練習

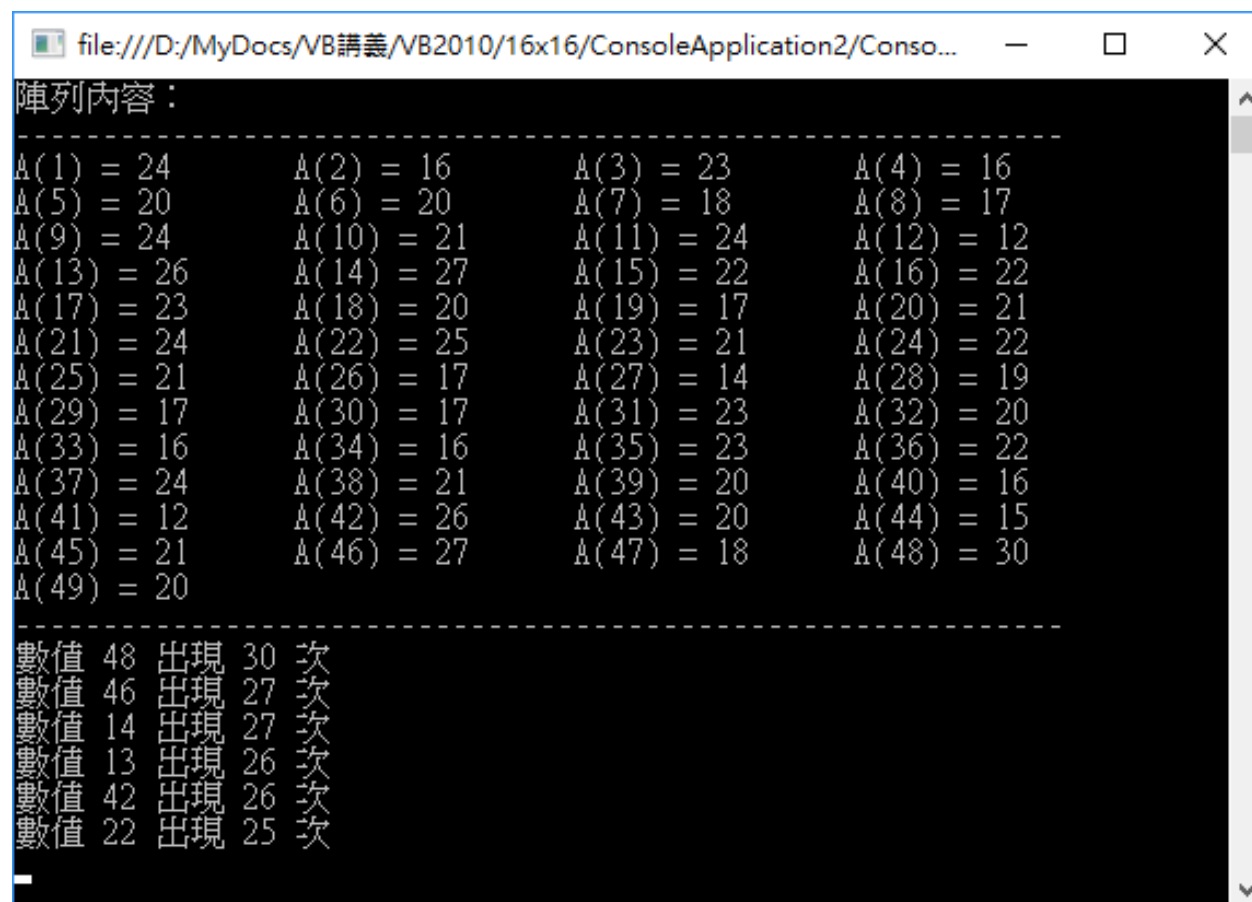
```
Randomize()  
Dim A(5), tmp As Integer  
For i = 0 To 5      '產生6個亂數  
    A(i) = Int(Rnd() * 49) + 1  
Next  
For i = 0 To UBound(A) - 1    '選擇排序  
    For j = i + 1 To UBound(A)  
        If A(i) > A(j) Then  
            tmp = A(i)  
            A(i) = A(j)  
            A(j) = tmp  
        End If  
    Next  
Next  
For i = 0 To UBound(A)    '印出陣列  
    Console.Write(A(i) & " ")  
Next  
Console.Read()
```

綜合練習題

- ▶ 寫一程式，以亂數產生1000個1~49的整數，並記錄在陣列中：
 - ▶ 一、印出每個數字出現的次數
 - ▶ 二、由小到大排序後印出出現次數最多的前六個數字及其次數

綜合練習題

▶ 參考執行結果：



file:///D:/MyDocs/VB講義/VB2010/16x16/ConsoleApplication2/Conso...

陣列內容：

A(1) = 24	A(2) = 16	A(3) = 23	A(4) = 16
A(5) = 20	A(6) = 20	A(7) = 18	A(8) = 17
A(9) = 24	A(10) = 21	A(11) = 24	A(12) = 12
A(13) = 26	A(14) = 27	A(15) = 22	A(16) = 22
A(17) = 23	A(18) = 20	A(19) = 17	A(20) = 21
A(21) = 24	A(22) = 25	A(23) = 21	A(24) = 22
A(25) = 21	A(26) = 17	A(27) = 14	A(28) = 19
A(29) = 17	A(30) = 17	A(31) = 23	A(32) = 20
A(33) = 16	A(34) = 16	A(35) = 23	A(36) = 22
A(37) = 24	A(38) = 21	A(39) = 20	A(40) = 16
A(41) = 12	A(42) = 26	A(43) = 20	A(44) = 15
A(45) = 21	A(46) = 27	A(47) = 18	A(48) = 30
A(49) = 20			

數值 48 出現 30 次
數值 46 出現 27 次
數值 14 出現 27 次
數值 13 出現 26 次
數值 42 出現 26 次
數值 22 出現 25 次

綜合練習題(參考寫法)

- ▶ 利用兩個陣列，一個紀錄每個數字出現的次數，一個紀錄排序的順序

A	0	16		30	20
	(0)	(1)		(48)	(49)

B	0	1		48	49
	(0)	(1)		(48)	(49)

綜合練習題(參考寫法)

```
Dim A(49), x As Integer
' 產生索引值陣列
Dim B(49), tmp As Integer
For i = 1 To 49
    B(i) = i
Next
' 產生1000個1~49的亂數，並將相對的陣列值+1
Randomize()
For i = 1 To 1000
    x = Int(Rnd() * 49) + 1
    A(x) = A(x) + 1
Next
```

綜合練習題(參考寫法)

· 印出陣列內容

```
Console.WriteLine( "陣列內容 : " )
Console.WriteLine("-----")
-----")
For i = 1 To 48 Step 4
    Console.WriteLine("A(" & i & ") = " & A(i) & vbCrLf &
        "A(" & i + 1 & ") = " & A(i + 1) & vbCrLf &
        "A(" & i + 2 & ") = " & A(i + 2) & vbCrLf &
        "A(" & i + 3 & ") = " & A(i + 3))
Next
Console.WriteLine("A(49) = " & A(49))
Console.WriteLine("-----")
-----")
```

綜合練習題(參考寫法)

'對陣列元素大小排序，但記錄其索引值

For i = 1 To 48 '使用選擇排序

For j = i + 1 To 49

If A(i) < A(j) Then

tmp = A(i) '交換A陣列兩元素

A(i) = A(j)

A(j) = tmp

tmp = B(i) '交換B陣列兩元素

B(i) = B(j)

B(j) = tmp

End If

Next

Next

綜合練習題(參考寫法)

· 印出排序後陣列前六個

```
For i = 1 To 6
    Console.WriteLine( "數值 " & B(i) & " 出現 "
                        & A(i) & " 次" )
Next

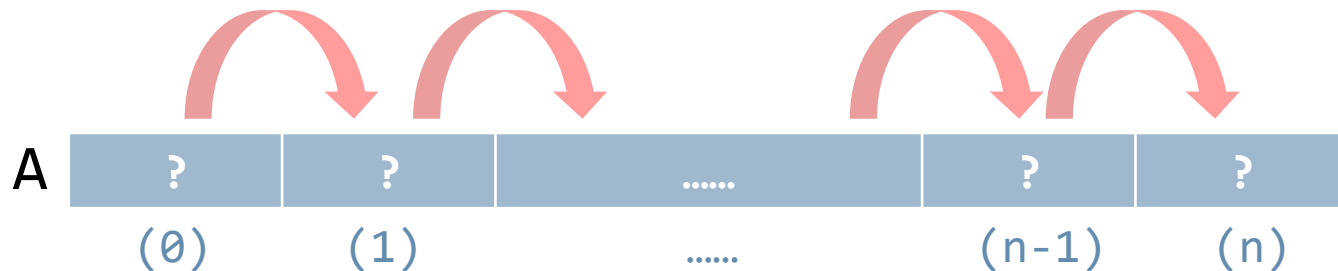
Console.Read()
```

搜尋(Search)

- ▶ 搜尋就是在一堆資料中找出所要之特定資料。
- ▶ 搜尋之主要核心動作為「比較」動作，必需透過比較才有辦法判斷是否尋找到特定資料。
- ▶ 資料未排序時，使用循序搜尋
- ▶ 排序過的資料，可使用二分搜尋或其他搜尋方式

循序搜尋

- ▶ 就是一個一個依序搜尋，效率較差，但資料若未經排序，僅能使用此方法
- ▶ 例如找某一值，則從 $A(0)$ 開始，一個個比較下去



循序搜尋練習

- 產生100個1~100的亂數，找出是否有使用者指定的數值

```
file:///D:/MyDocs/VB講義/VB2010/16x16/ConsoleApplic...
1=59,2=6,3=56,4=77,5=68,6=33,7=40,8=54,9=78,10=99,11=74,12=
75,13=64,14=13,15=42,16=75,17=82,18=39,19=51,20=10,21=12,22
=66,23=42,24=41,25=31,26=95,27=21,28=31,29=8,30=51,31=18,32
=71,33=23,34=25,35=68,36=35,37=13,38=34,39=22,40=52,41=45,4
2=15,43=78,44=13,45=5,46=36,47=37,48=91,49=24,50=36,51=89,5
2=1,53=21,54=89,55=20,56=68,57=75,58=93,59=41,60=33,61=17,6
2=82,63=53,64=80,65=53,66=71,67=24,68=85,69=9,70=12,71=98,7
2=95,73=1,74=89,75=31,76=30,77=30,78=21,79=12,80=5,81=2,82
50,83=5,84=86,85=75,86=4,87=48,88=66,89=1,90=1,91=1,92=1,93
6,93=55,94=49,95=6,96=61,97=89,98=22,99=1,100=1
```

請輸入要搜尋的數值(1~100)：41
在位置A(24)找到 41

```
file:///D:/MyDocs/VB講義/VB2010/16x16/ConsoleApplica...
1=40,2=83,3=62,4=42,5=23,6=12,7=67,8=79,9=79,10=67,11=60,12=
52,13=37,14=88,15=28,16=74,17=11,18=73,19=87,20=86,21=35,22=
41,23=82,24=29,25=45,26=23,27=27,28=46,29=65,30=76,31=57,32=
11,33=30,34=22,35=15,36=78,37=36,38=10,39=56,40=58,41=4,42=9
,43=86,44=22,45=76,46=16,47=11,48=28,49=42,50=3,51=28,52=67,
53=75,54=74,55=29,56=48,57=9,58=59,59=32,60=94,61=34,62=24,6
3=43,64=69,65=12,66=13,67=36,68=30,69=24,70=11,71=64,72=7,73
=43,74=34,75=85,76=99,77=25,78=41,79=29,80=3,81=34,82=74,83=
72,84=67,85=83,86=24,87=51,88=65,89=8,90=18,91=92,92=99,93=5
9,94=32,95=71,96=24,97=25,98=34,99=93,100=3,
請輸入要搜尋的數值(1~100)：100
沒有找到 100
```

循序搜尋練習

▶ 參考寫法：

```
Dim A(100), x As Integer
Dim find As Boolean
Randomize()
For i = 1 To 100    '產生100個亂數
    A(i) = Int(Rnd() * 100) + 1
    Console.Write(i & "=" & A(i) & ",")
Next
Console.WriteLine()
Console.Write("請輸入要搜尋的數值(1~100) : ")
x = Console.ReadLine()    '請求輸入
```

循序搜尋練習

```
find = False
For i = 1 To 100    '依序比較每一個元素
    If A(i) = x Then
        Console.WriteLine("在位置A(" & i & ")找到 " & x)
        find = True
        Exit For    '找到就印出並離開迴圈
    End If
Next
If Not find Then '如果變數find沒有被設定成True，表示未找到
    Console.WriteLine("沒有找到 " & x)
End If
Console.Read()
```

循序搜尋練習

- ▶ 產生10000個1~10000的亂數，找出是否有使用者指定的數值
- ▶ 顯示搜尋次數及位置



```
file:///D:/MyDocs/VB講義/VB2010/16...  
請輸入要搜尋的數值(1~10000): 500  
共搜尋 1595次  
在位置A(1595)找到 500  
  
(找了1595次)
```



```
file:///D:/MyDocs/VB講義/V...  
請輸入要搜尋的數值(1~10000): 3621  
共搜尋10000次  
沒有找到 3621  
  
(找了10000次)
```

循序搜尋練習

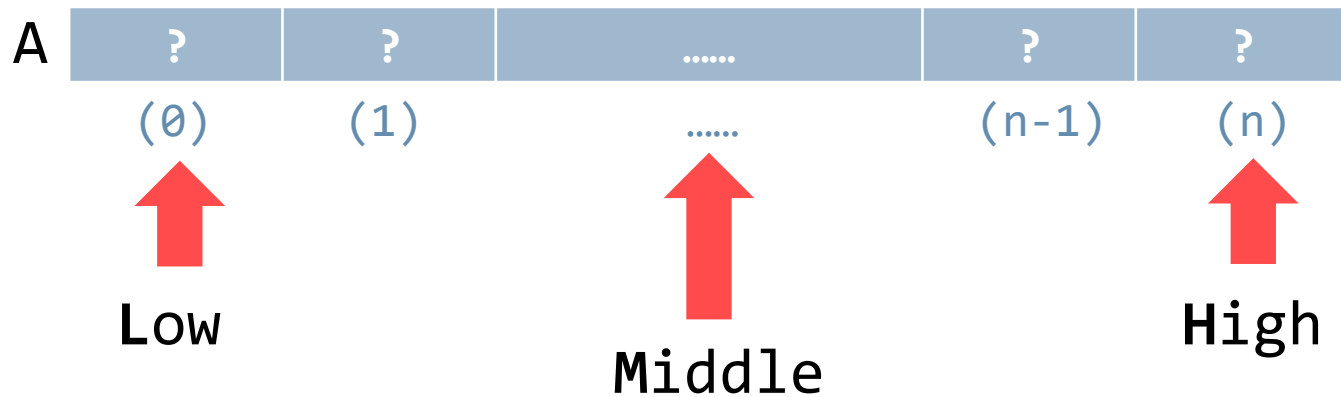
```
Dim A(10000), x As Integer
Dim find As Boolean
Randomize()
For i = 1 To 10000    '產生亂數
    A(i) = Int(Rnd() * 10000) + 1
Next
Console.WriteLine("請輸入要搜尋的數值(1~10000) : ")
x = Console.ReadLine()
find = False
For i = 1 To 10000    '從第一個元素開始依序向後搜尋
    If A(i) = x Then
        Console.WriteLine("共搜尋 " & i & "次")
        Console.WriteLine("在位置A(" & i & ")找到 " & x)
        find = True
        Exit For
    End If
Next
If Not find Then
    Console.WriteLine("共搜尋10000次")
    Console.WriteLine("沒有找到 " & x)
End If
Console.Read()
```

循序搜尋

- ▶ 循序搜尋特性：
 - ▶ 資料不需事先排序
 - ▶ N個資料，則搜尋比對的次數最少1次，最多N次
 - ▶ 平均需要比對 $(N+1)/2$ 次

二分搜尋(Binary Search)

- ▶ 必要條件：資料必須經過排序
- ▶ 方法：每次將資料分為兩半，看指到的資料是否符合，若不是，則再搜尋符合範圍內的一半資料即可

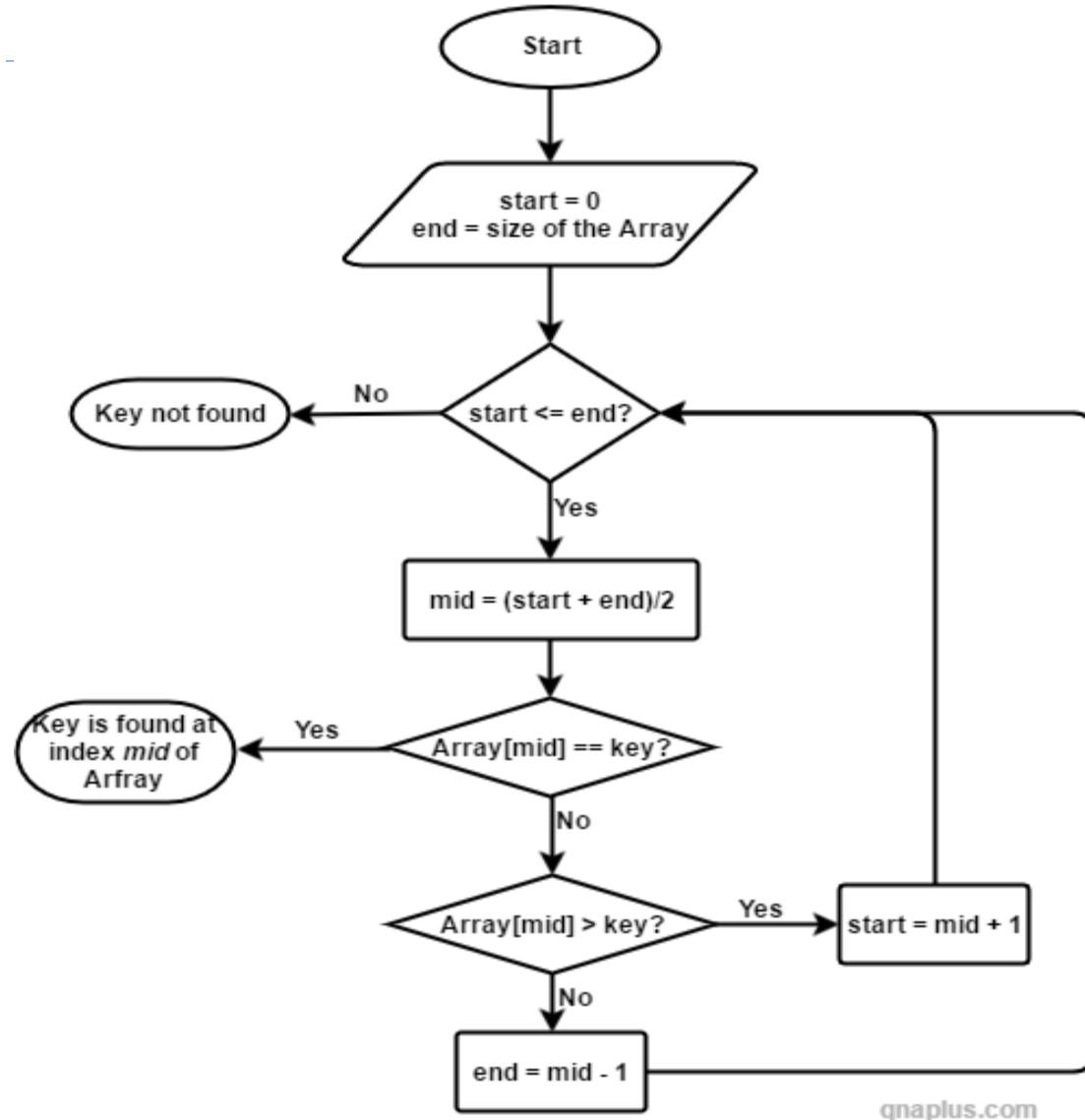


二分搜尋

- ▶ 設有一已排序陣列A，則：
- ▶ Low， $L =$ 第一個元素， $A(0)$
- ▶ High， $H =$ 最後一個元素， $UBound(A)$
- ▶ Middle， $M =$ 中間值， $(L + H) / 2$
- ▶ 若 $A(M)$ 即為要搜尋的值，則結束
- ▶ 若要搜尋的值 $> A(M)$ ，則 $L = M + 1$
- ▶ 若要搜尋的值 $< A(M)$ ，則 $H = M - 1$
- ▶ 重複此動作直到 $H < L$ ，則表示搜尋失敗，找不到

二分搜尋

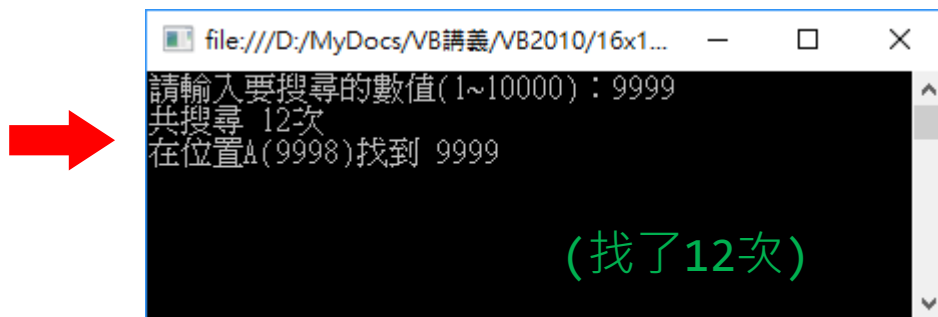
Binary search algorithm: find *key* in a sorted *Array*



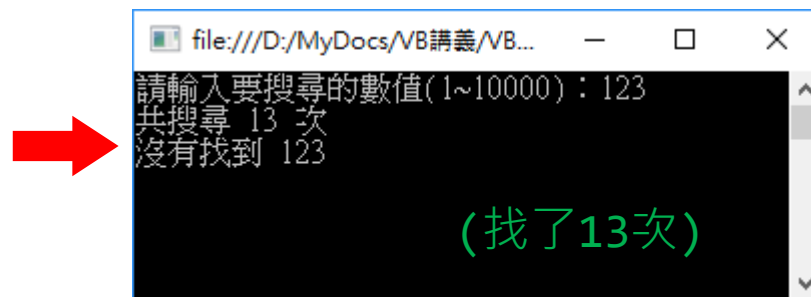
qnaplus.com

二分搜尋練習

- ▶ 產生10000個1~10000的亂數，找出是否有使用者指定的數值
- ▶ 顯示搜尋次數及位置



```
file:///D:/MyDocs/VB講義/VB2010/16x1...  
請輸入要搜尋的數值(1~10000): 9999  
共搜尋 12次  
在位置A(9998)找到 9999  
  
(找了12次)
```



```
file:///D:/MyDocs/VB講義/VB...  
請輸入要搜尋的數值(1~10000): 123  
共搜尋 13 次  
沒有找到 123  
  
(找了13次)
```

二分搜尋練習

```
Dim A(10000), x, L, H, M, num, tmp As Integer
Dim find As Boolean
Randomize()
For i = 1 To 10000      '產生亂數
    A(i) = Int(Rnd() * 10000) + 1
Next
For i = 2 To UBound(A)  '氣泡排序
    For j = 1 To UBound(A) - i
        If A(j) > A(j + 1) Then
            tmp = A(j)
            A(j) = A(j + 1)
            A(j + 1) = tmp
        End If
    Next
Next
Console.Write("請輸入要搜尋的數值(1~10000) : ")
x = Console.ReadLine()  '請求輸入
```

要先排序

二分搜尋練習

find = False

‘設定變數初值

L = 1

H = UBound(A)

Do

‘執行二分搜尋

num += 1

M = (L + H) \ 2

If A(M) = x Then

find = True

Exit Do

ElseIf X < A(M) Then

H = M - 1

Else

L = M + 1

End If

Loop Until H < L

二分搜尋

二分搜尋練習

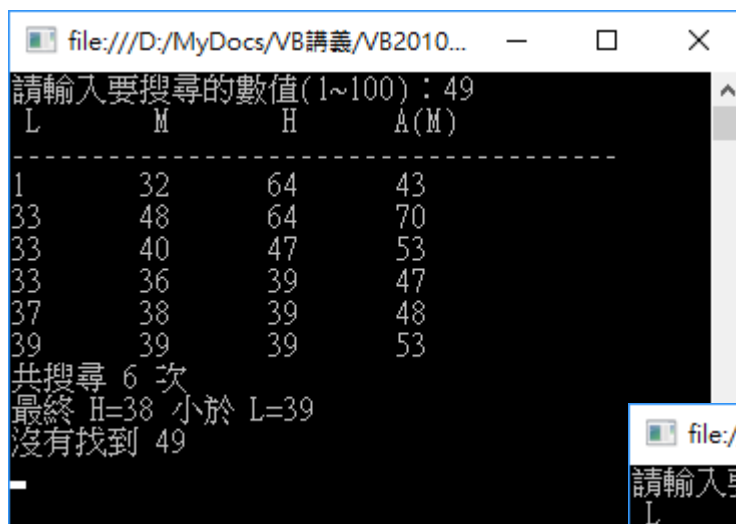
```
▶ Console.WriteLine("共搜尋 " & num & " 次")  
If find Then  
    Console.WriteLine("在位置A(" & M & ")找到 " & x)  
else  
    Console.WriteLine("沒有找到 " & x)  
End If  
Console.Read()
```

二分搜尋

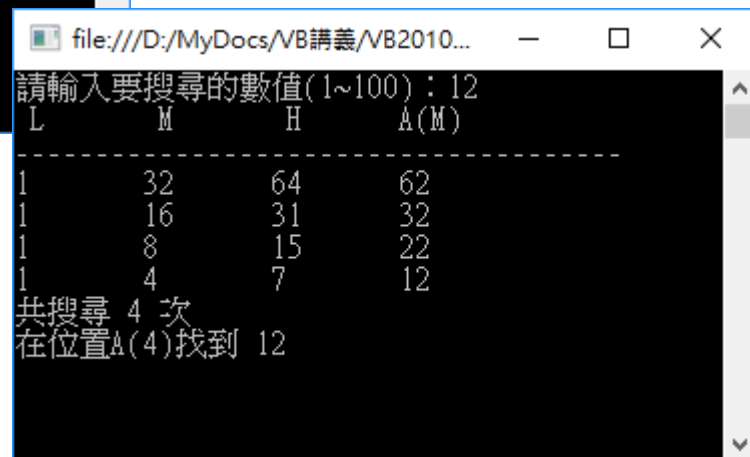
- ▶ 二分搜尋特性：
 - ▶ 資料必需事先排序
 - ▶ N個資料，則搜尋比對的次數最少1次，最多 $\log_2 N + 1$ 次
- ▶ 說明：
 - ▶ 若有64筆資料，則搜尋次數最多為 $\log_2 64 = 6$ (找到)，若找不到則多加一次 = 7次
 - ▶ 若Log不好算，換一個說法： 2^X 必須 ≥ 64 ，故 $X=6$

二分搜尋練習

- 寫一程式，產生64個1~100的亂數，排序後以二分搜尋找尋指定的值，要顯示搜尋過程中L、H、M及A(M)的值。



```
file:///D:/MyDocs/VB講義/VB2010...
請輸入要搜尋的數值(1~100): 49
L      M      H      A(M)
-----
1      32      64      43
33     48      64      70
33     40      47      53
33     36      39      47
37     38      39      48
39     39      39      53
共搜尋 6 次
最終 H=38 小於 L=39
沒有找到 49
```



```
file:///D:/MyDocs/VB講義/VB2010...
請輸入要搜尋的數值(1~100): 12
L      M      H      A(M)
-----
1      32      64      62
1      16      31      32
1      8       15      22
1      4       7       12
共搜尋 4 次
在位置A(4)找到 12
```

二分搜尋練習

▶ 參考寫法：

```
Dim A(64), x, L, H, M, num, tmp As Integer
Dim find As Boolean
Randomize()
For i = 1 To 64      '產生亂數
    A(i) = Int(Rnd() * 100) + 1
Next
For i = 2 To UBound(A)    '氣泡排序
    For j = 1 To UBound(A) - i
        If A(j) > A(j + 1) Then
            tmp = A(j)
            A(j) = A(j + 1)
            A(j + 1) = tmp
        End If
    Next
Next
```

二分搜尋練習



```
Console.Write("請輸入要搜尋的數值(1~100) : ")
x = Console.ReadLine()      '請求輸入
Console.WriteLine("  L          M          H          A(M) ")
Console.WriteLine("-----")
find = False                 '設定變數初值
L = 1
H = UBound(A)
```



二分搜尋練習

```
Do      '執行二分搜尋
    num += 1
    M = (L + H) \ 2
    Console.WriteLine(L & vbTab & M & vbTab &
                       H & vbTab & A(M))

    If A(M) = x Then
        find = True
        Exit Do
    ElseIf x < A(M) Then
        H = M - 1
    Else
        L = M + 1
    End If
Loop Until H < L
```

二分搜尋練習



```
Console.WriteLine("共搜尋 " & num & " 次")
If find Then
    Console.WriteLine("在位置A(" & M & ")找到 " & x)
Else
    Console.WriteLine("最終 H=" & H & " 小於 L=" & L)
    Console.WriteLine("沒有找到 " & x)
End If
Console.Read()
```



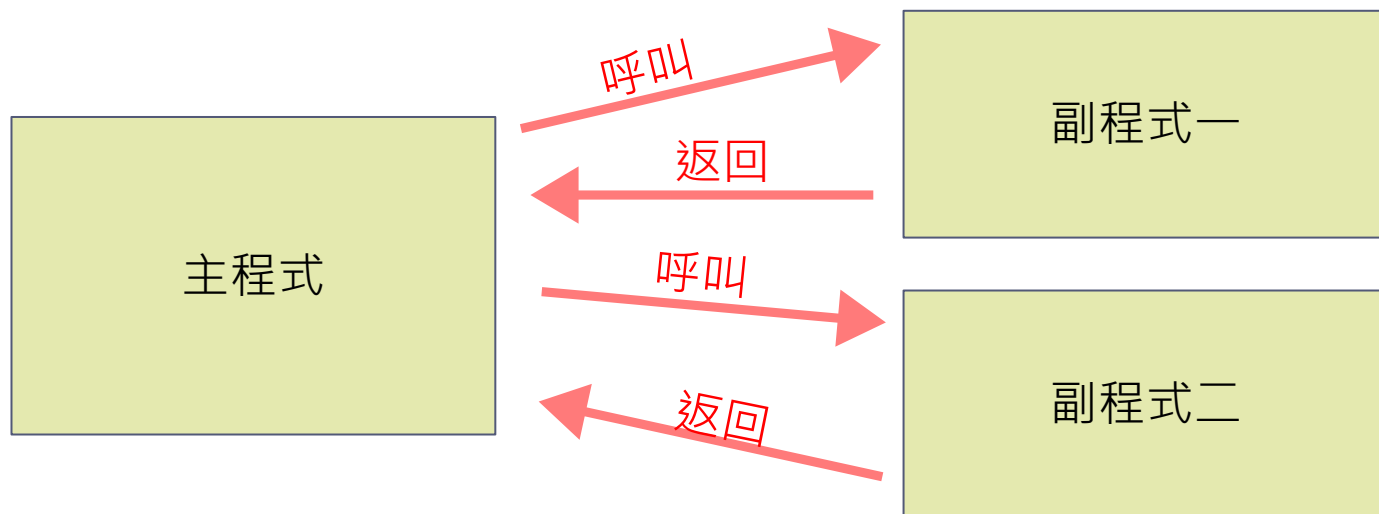
程序(副程式)與函數



编号: 732913 (图片114网 www.tupian114.com) 永久免费素材网

程序(副程式)與函數

- ▶ 當程式中出現相同的段落時，應寫成副程式的方式，再呼叫即可，可簡化程式，減少出錯及維護難度
- ▶ 若之後可能用到相同的功能，也可寫成副程式或函數，供之後呼叫
- ▶ 利用副程式與函數達成模組化



程序(副程式)與函數

- ▶ 副程式與函數最大的差別：

副程式	沒有	傳回值
函數	一定有	傳回值

- ▶ 所以呼叫函數需準備變數接收傳回值

副程式

▶ 語法一(沒有參數)：

```
Sub 副程式名稱()  
    :  
    :  
End Sub
```

▶ 語法二(有參數)

```
Sub 副程式名稱(參數一,參數二,.....)  
    :  
    :  
End Sub
```

副程式(沒有傳遞參數)

► 例：

要印出陣列內容兩次

Call指令
可省略

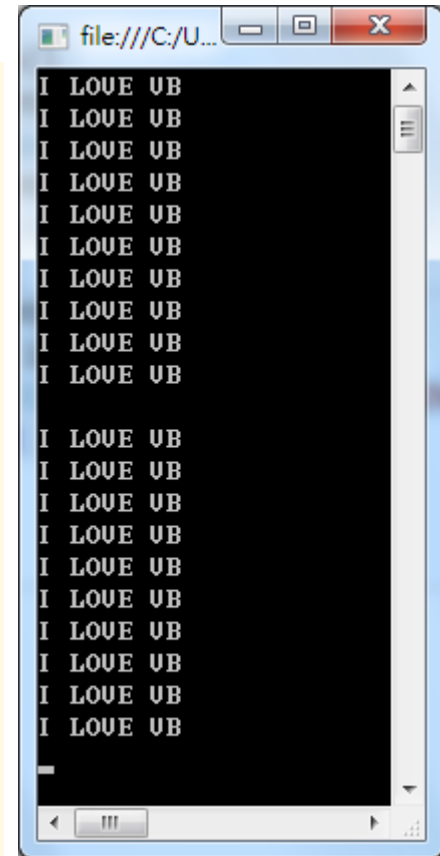
寫成副程式再呼叫

```
Dim A(10) As Integer
Sub Main()
    For i = 1 To 10 '亂數產生數值
        A(i) = Int(Rnd() * 10) + 1
    Next
    Call PrintRnd() '印出陣列內容
    For i = 1 To 10
        A(i) *= 10 '個元素*10
    Next
    PrintRnd() '印出陣列內容
    Console.Read()
End Sub
Sub PrintRnd()
    For i = 1 To 10
        Console.Write(A(i) & ",")
    Next
    Console.WriteLine()
End Sub
```

副程式(沒有傳遞參數)

```
Sub Main()  
    printA()  
    Console.WriteLine()  
    printA()  
    Console.Read()  
End Sub
```

```
Sub printA()  
    For i = 1 To 10  
        Console.WriteLine("I LOVE VB")  
    Next  
End Sub
```

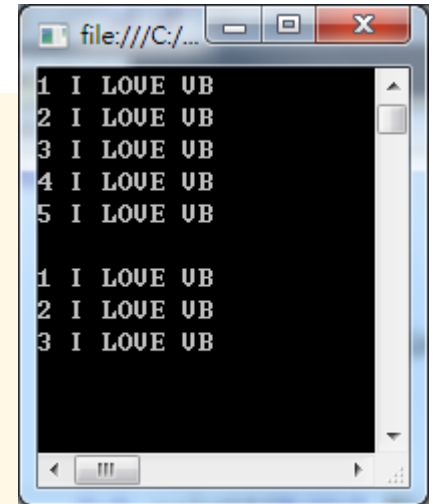


副程式(有傳遞參數)

- ▶ 將數值傳給副程式的X：

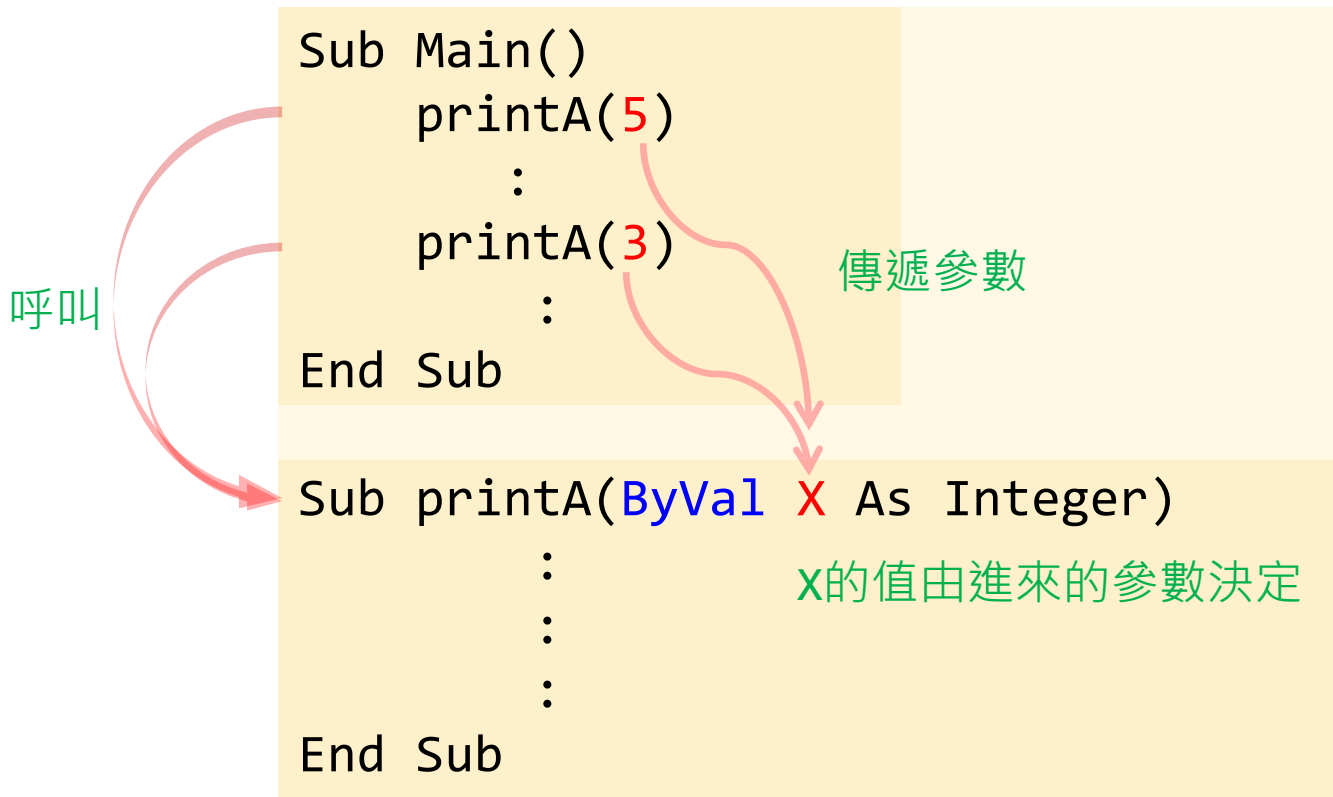
```
Sub Main()  
    printA(5)  
    Console.WriteLine()  
    printA(3)  
    Console.Read()  
End Sub
```

```
Sub printA(ByVal X As Integer)  
    For i = 1 To X  
        Console.WriteLine(i & " I LOVE VB")  
    Next  
End Sub
```



副程式(有傳遞參數)

- ▶ 呼叫副程式時將參數傳遞給副程式的變數X接收



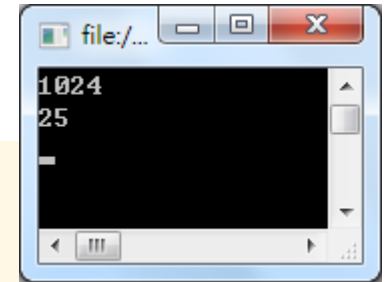
副程式(傳遞多個參數)

▶ 參數之間用逗點隔開

計算m的n次方

```
Sub Main()  
    printA(2, 10)  
    printA(5, 2)  
    Console.Read()  
End Sub
```

```
Sub printA(ByVal X As Integer, ByVal Y As Integer)  
    Console.WriteLine(X ^ Y)  
End Sub
```



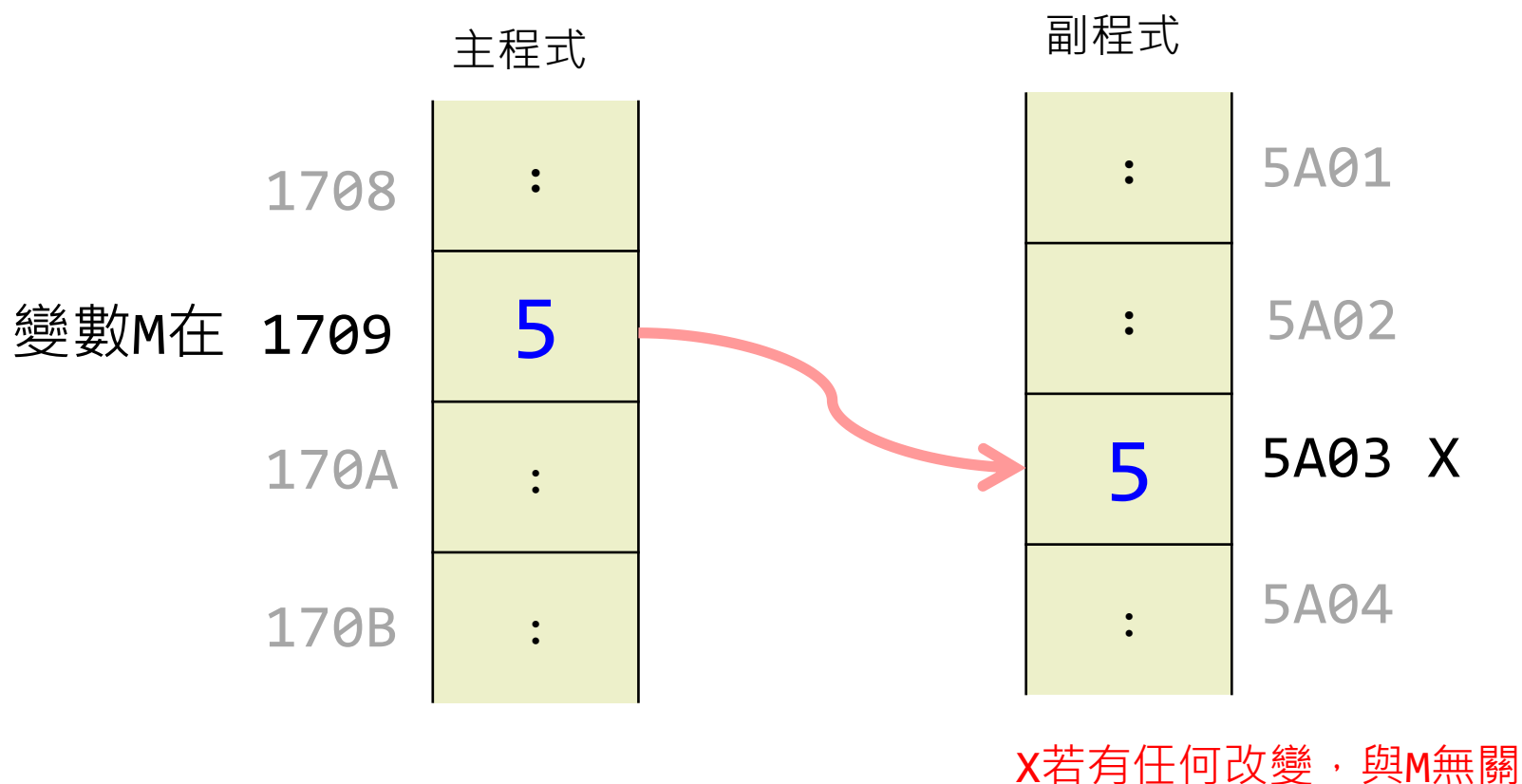
ByVal 與 ByRef

- ▶ ByVal：傳值，將參數的**值**傳給副程式或函數
- ▶ ByRef：傳址，將參數的記憶體**位址**傳給副程式或函數
- ▶ 這是一般變數宣告後的狀況：



ByVal 與 ByRef

- ▶ ByVal：將值傳過去



ByVal 與 ByRef

- ▶ ByRef：將**記憶體位址**傳過去，基本上就是與原變數使用同一個位址，即兩個變數都指到同一個地方

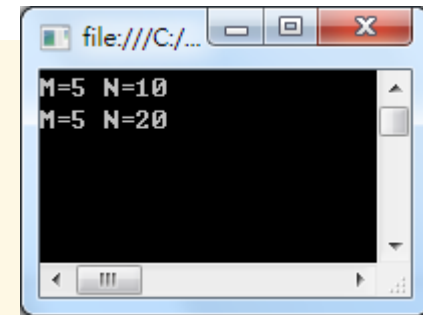


ByVal 與 ByRef

- ▶ M使用ByVal，N使用ByRef，可以看出N被副程式改變了

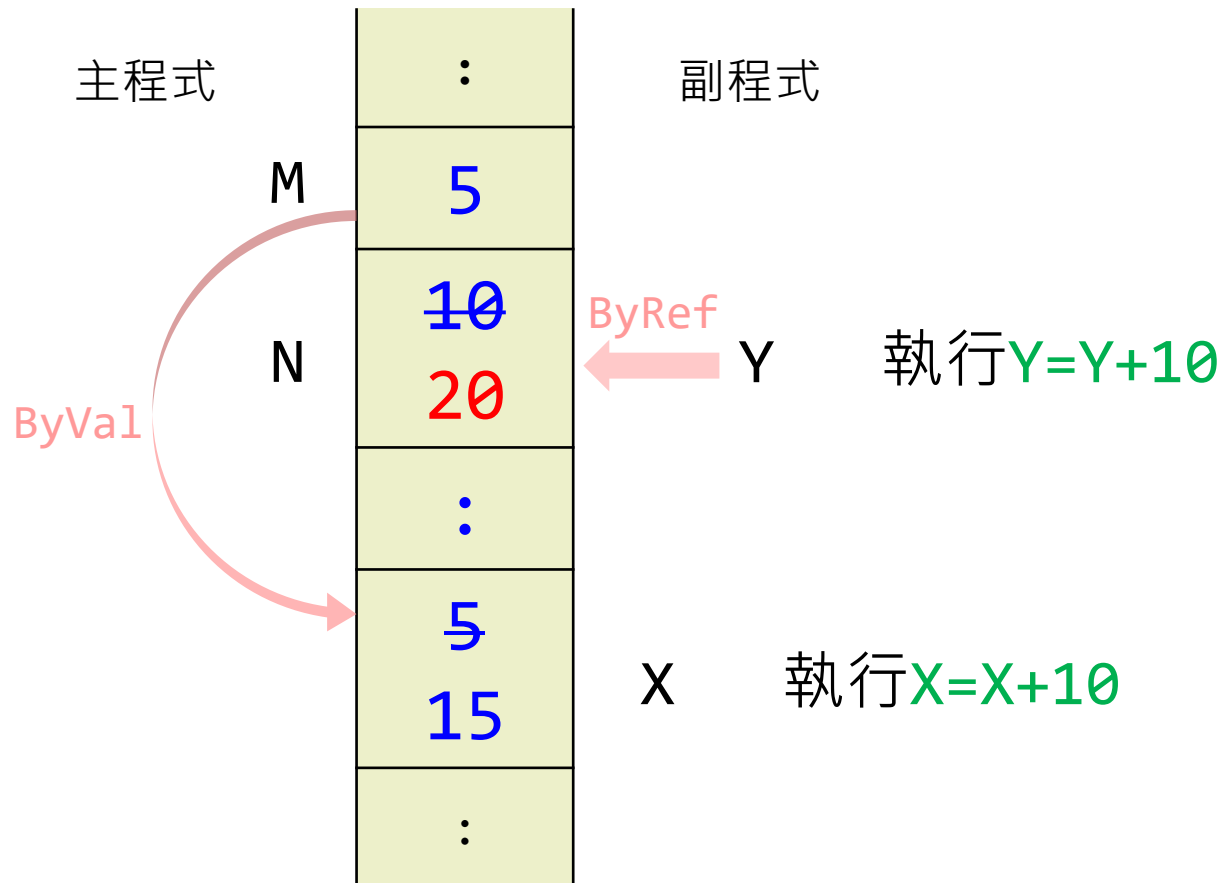
```
Sub Main()  
    Dim M, N As Integer  
    M = 5 : N = 10  
    Console.WriteLine("M=" & M & " N=" & N)  
    printA(M, N)  
    Console.WriteLine("M=" & M & " N=" & N)  
    Console.Read()  
End Sub
```

```
Sub printA(ByVal X As Integer, ByRef Y As Integer)  
    X = X + 10  
    Y = Y + 10  
End Sub
```



ByVal 與 ByRef

- ▶ 前例記憶體使用情形：



ByVal 與 ByRef

- ▶ 若原變數須被副程式改變，就使用ByRef，否則使用ByVal，依程式需求使用
- ▶ ByRef也相當於副程式將執行結果傳回給主程式
- ▶ 若未宣告ByVal或ByRef，則視為ByVal
- ▶ 若未宣告參數型態，則視為與傳入型態相同

函數(Function)

► 語法：

```
Function 函數名稱(參數一, 參數二, .....)  
    :  
    :  
    Return 傳回值  
End Function
```

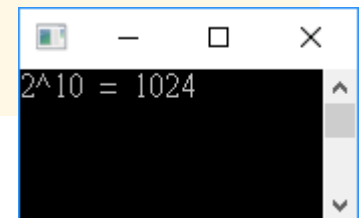
- 與副程式相同，參數可有可無
- 函數中執行到Return敘述則結束函數，並將指定的值傳回

函數(Function)

▶ 有傳遞參數：

```
Sub Main()  
    Dim a, b, x As Integer  
    a = 2 : b = 10  
    x = sqr(a, b) '要用變數接收傳回值  
    Console.WriteLine(a & "^" & b & " = " & x)  
    Console.Read()  
End Sub
```

```
Function sqr(x, y)  
    Return x ^ y  
End Function
```

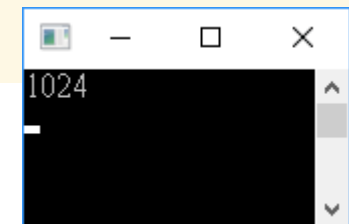


函數(Function)

► 簡化版：

```
Sub Main()  
    '直接使用傳回值  
    Console.WriteLine(sqr(2, 10))  
    Console.Read()  
End Sub
```

```
Function sqr(x, y)  
    Return x ^ y  
End Function
```



函數(Function)

- ▶ 除了可利用Return傳回值外，也可利用與函數名稱相同之變數名稱來傳回值，結果是相同的。
- ▶ 例：用Return

```
Function sqr(x, y)  
    Return x ^ y  
End Function
```

- ▶ 例：用函數名稱

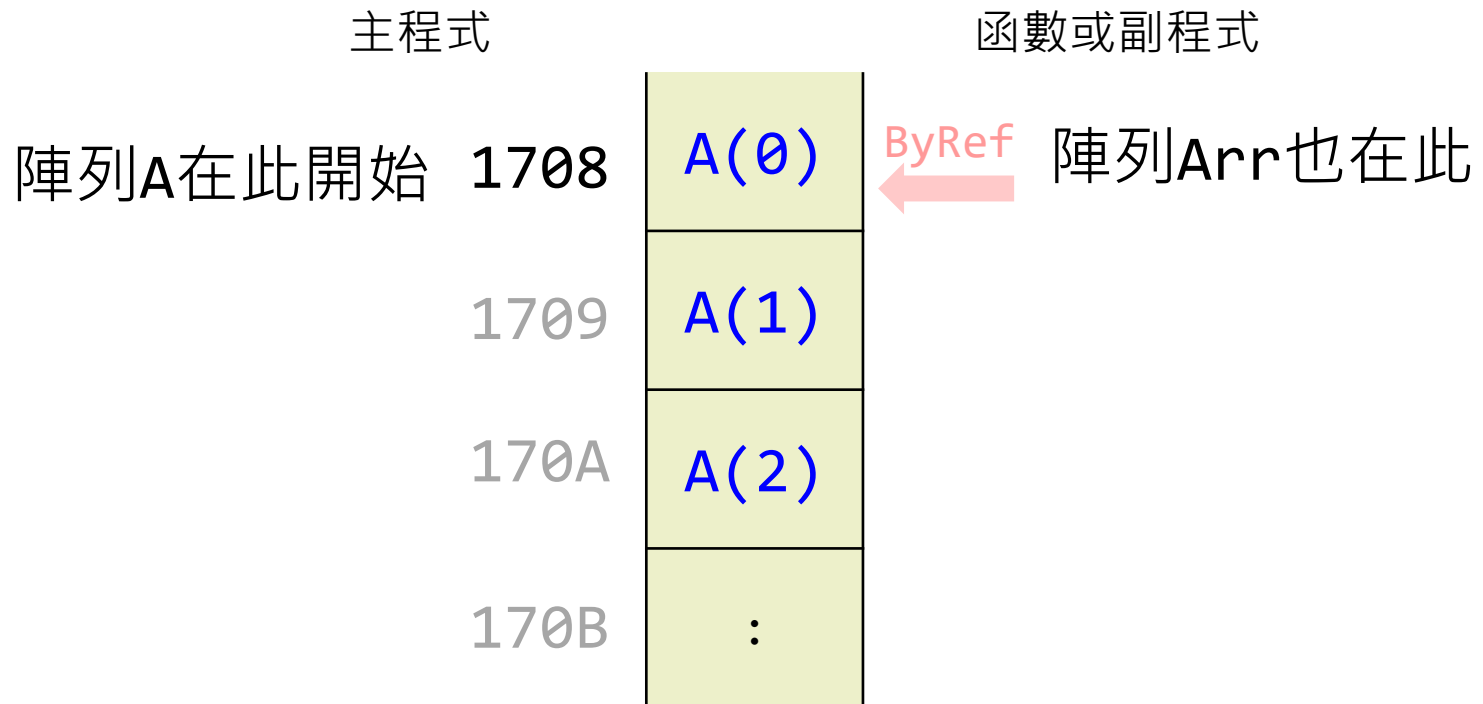
```
Function sqr(x, y)  
    sqr = x ^ y  
End Function
```

傳遞陣列

- ▶ 若傳遞單一元素，例如A(1)，則跟傳遞變數相同
- ▶ 若要傳遞一整個陣列，則使用陣列名稱即可，且不管宣告為ByVal或ByRef，一律視為ByRef

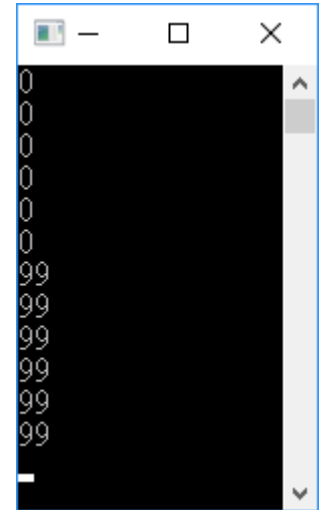
傳遞陣列

- ▶ 是將陣列的**記憶體位址**傳過去，基本上就是與原陣列使用同一個位址，即兩個陣列都指到同一個地方



傳遞陣列

```
Sub Main()  
    Dim A(5) As Integer  
    PrintArr(A) '印出陣列  
    SetTo99(A) '將所有元素值設為99  
    PrintArr(A) '印出陣列  
    Console.Read()  
End Sub  
  
Sub PrintArr(ByRef Arr() As Integer)  
    For i = 0 To UBound(Arr)  
        Console.WriteLine(Arr(i))  
    Next  
End Sub  
  
Sub SetTo99(ByRef Arr() As Integer)  
    For i = 0 To UBound(Arr)  
        Arr(i) = 99  
    Next  
End Sub
```



跳出副程式或函數

- ▶ 當想立即結束副程式或函數時，使用下列指令：
- ▶ 跳出副程式：

Exit Sub

- ▶ 跳出函數：

Exit Function

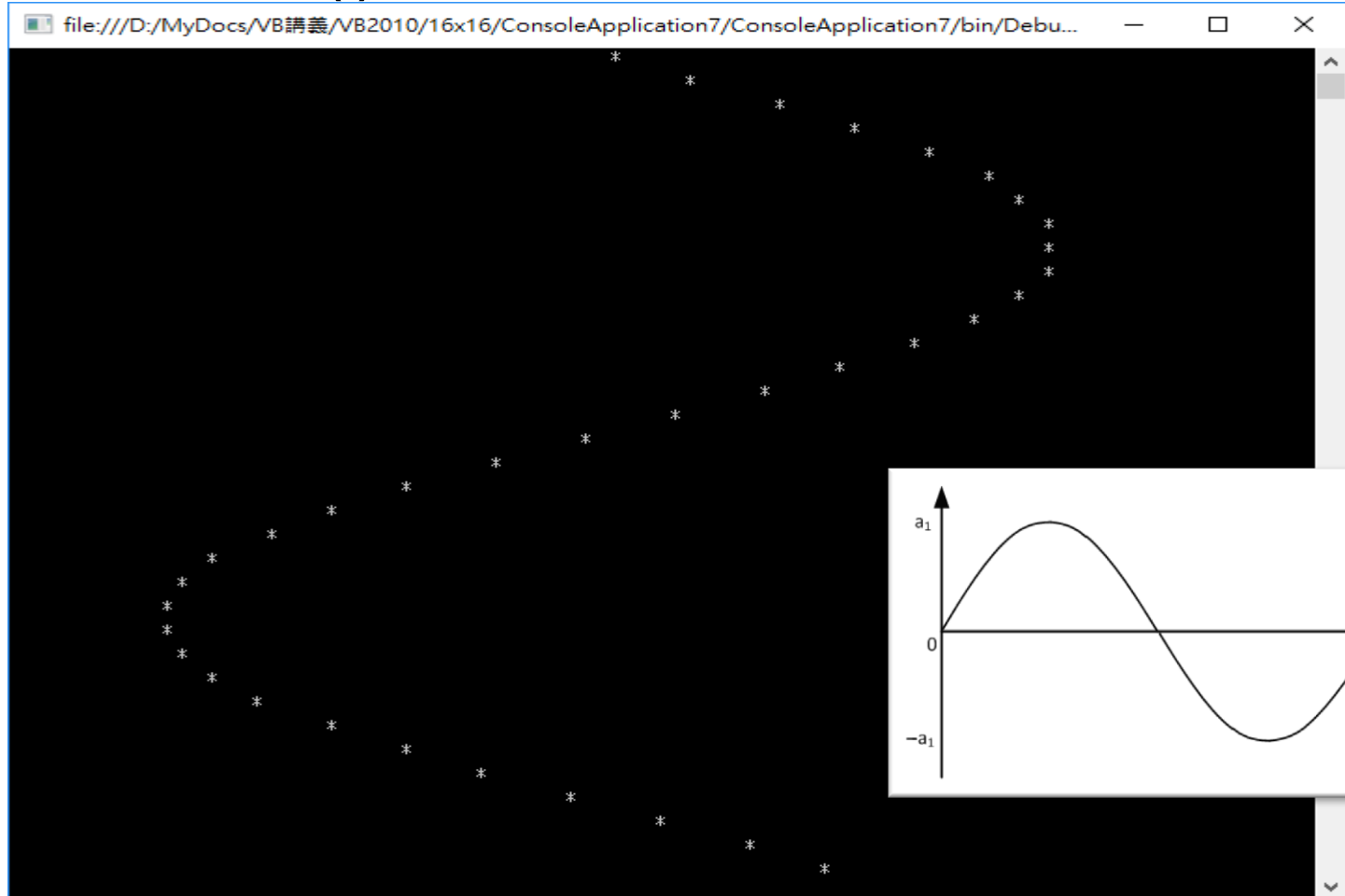
- ▶ 執行上述敘述後，控制權及交還主程式

內建函數

- ▶ 所有的程式語言都有事先準備好一堆函數以供呼叫，可大幅減少程式開發的負擔，要好好利用
- ▶ VB有提供：
 - ▶ 字串函數
 - ▶ 數學及亂數函數
 - ▶ 日期與時間函數
 - ▶ 資料轉換函數
 - ▶ 資料格式化函數
- ▶ 請參看書籍或手冊

內建函數範例

- ▶ 利用`Math.Sin()`函數畫一個正弦波



內建函數範例

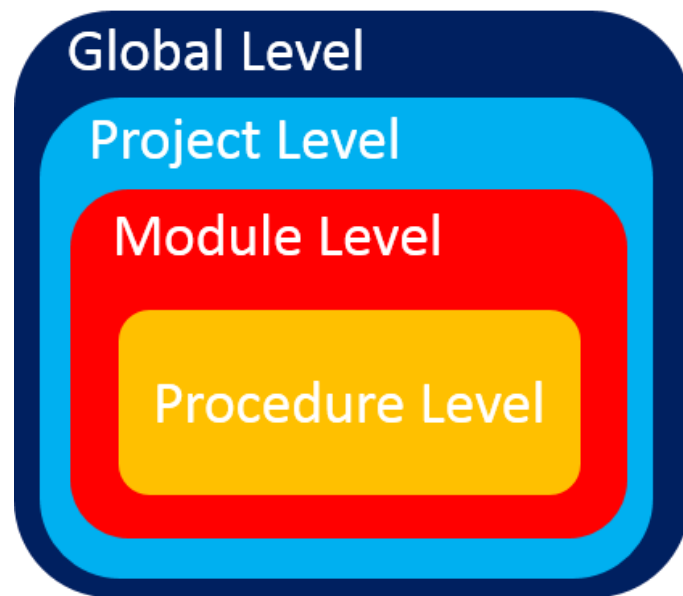
▶ 參考寫法：

```
Sub Main()  
    Dim x As Double  
    For i = 0 To 30 Step 0.2  
        x = Int( Math.Sin(i) * 30)  
        Console.WriteLine(Space(40 + x) & "*")  
    Next  
    Console.Read()  
End Sub
```

▶ 請自行試試其他內定函數

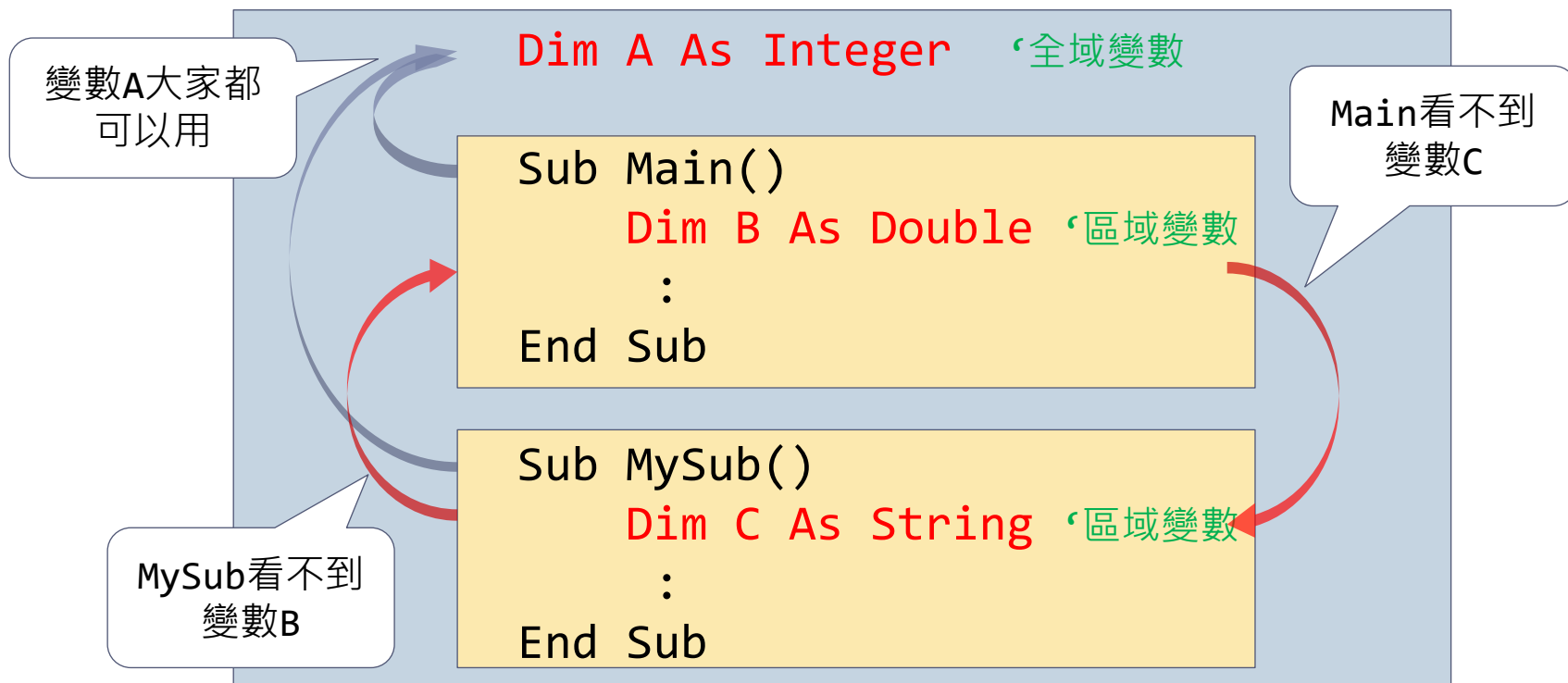
變數的可見範圍及生命週期

- ▶ 變數依其宣告時的位置及方式，決定了它的**可見範圍**(誰可以看到並使用它)及**生命週期**(在記憶體內停留的時間)
- ▶ 目的是在保護變數，避免不當的存取
- ▶ 區分為：
 - ▶ 全域變數
 - ▶ 區域變數
 - ▶ 其他



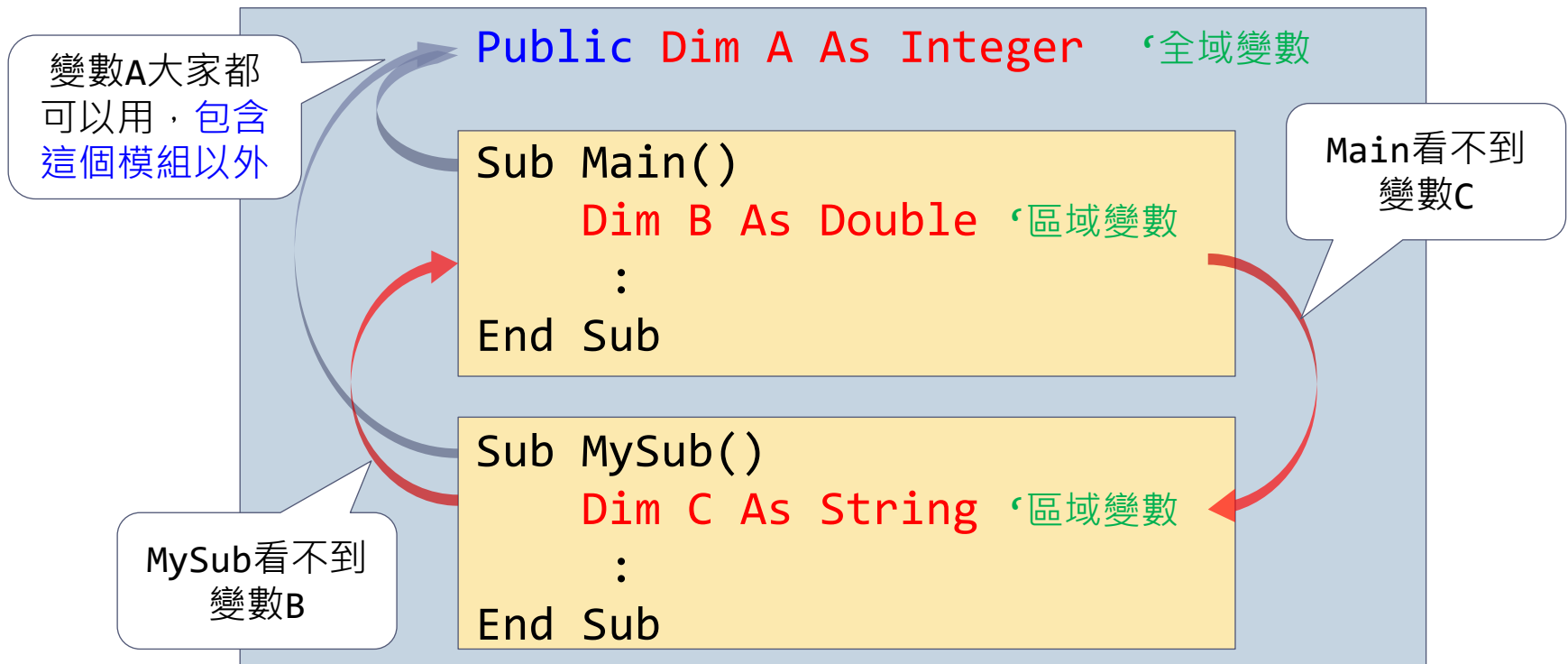
變數的可見範圍

- ▶ 全域變數：
 - ▶ 1.宣告在所有段落之外的變數
 - ▶ 2.Public修飾字可讓變數跨模組



變數的可見範圍

- ▶ 全域變數：
 - ▶ 1.宣告在所有段落之外的變數
 - ▶ 2.Public修飾字可讓變數跨模組



變數的生命週期

- ▶ 當變數被配置了記憶體，它就「出生」了，當它不再占用記憶體，它的生命就結束了，稱之為變數的「**生命週期**」
- ▶ **全域變數**一經宣告，整個程式執行的過程中都存在於記憶體中，直到程式結束
- ▶ **區域變數**只有該模組或副程式(函數)被呼叫執行時才占用記憶體，該段落結束時則消失

變數的可見範圍及生命週期

Module1.vb* x

Module1 (宣告)

```
Module Module1
    Dim A As Integer
    Sub Main()
        Dim B As Double
        A = 10
        B = 20
        C = 30
    End Sub
    Sub MySub()
        Dim C As String
        A = 10
        B = 20
        C = 30
    End Sub
End Module
```

錯誤！在Main裡看不到變數C

錯誤！在MySub裡看不到變數B

錯誤清單

2 項錯誤 0 項警告 0 項訊息

	描述	檔案	行	欄	專案
1	'C' 未宣告。由於其保護層級，可能無法對其進行存取。	Module1.vb	7	9	ConsoleApplication1
2	'B' 未宣告。由於其保護層級，可能無法對其進行存取。	Module1.vb	12	9	ConsoleApplication1

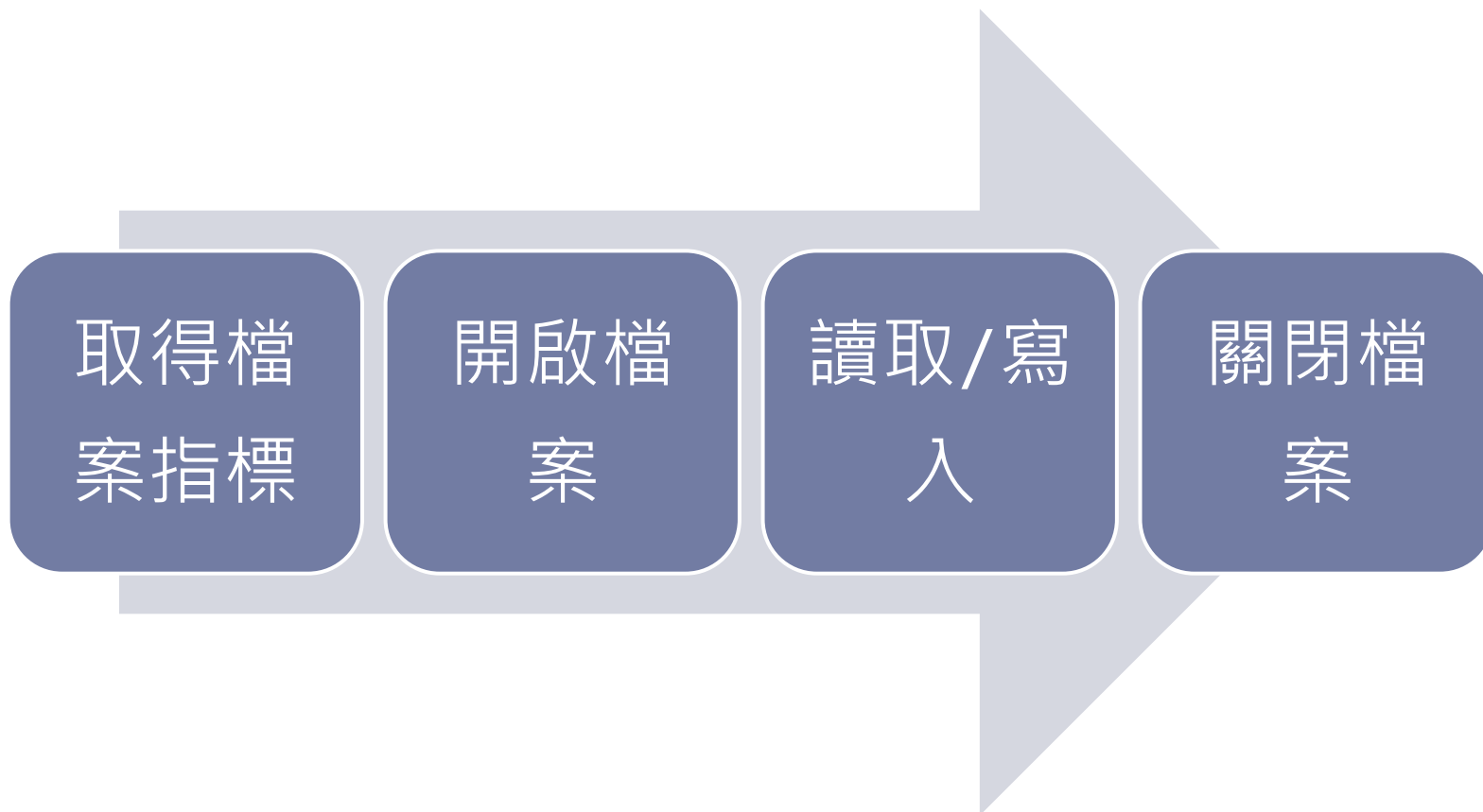
就緒

檔案讀寫



檔案讀寫

- ▶ 文字檔(.txt)讀取與寫入：



檔案讀寫

▶ 取得檔案指標

```
Dim fp As Integer  
:  
fp = FreeFile()
```

- ▶ 在程式中要開啟一個檔案必須先取得一個指標代碼，然後用此代碼在程式中代表這個檔案
- ▶ `FreeFile()` 函數會自動取得可用的代碼
- ▶ `fp` 是個整數變數，名稱依變數命名規則即可

檔案讀寫

▶ 開啟檔案

```
Dim fname As String  
fname = "d:\aaa.txt"    ‘要開啟的檔案  
:  
:  
FileOpen(fp, fname, OpenMode.Output)
```

檔案指標

檔案名稱

檔案開啟模式

▶ 注意必須有磁碟機及檔案存取的權限

檔案讀寫

▶ 檔案開啟模式

模 式	說 明
Output	開啟檔案進行寫入存取權。
Input	開啟檔案進行讀取存取權。
Append	若要將附加至開啟的檔案。（預設值）
Binary	二進位存取所開啟的檔案。
Random	開啟檔案進行隨機存取。

- ▶ Input、Output和 Append 時，會使用循序存取檔案，例如文字檔，而 Binary 用於二進位檔案存取，Random 用於隨機檔案存取
- ▶ 當循序存取檔案時，無法變更其資料。只可以讀取資料、附加到檔案，或以新的資料覆寫它。

檔案讀寫

▶ 寫入檔案：

寫入檔案，並加入換行符號

```
WriteLine(fp, "要寫入的文字")  
Write(fp, "要寫入的文字")
```

寫入檔案，沒有換行符號

▶ 讀取檔案：

```
Input(fp, 變數)
```

讀取的文字會放入變數中，每次讀取一行或逗點分隔之文字段落

若某一行之內容為：
“AAA”， “BBB”， “CCC”
則一次Input只讀入“AAA”

檔案讀寫

▶ 關閉檔案

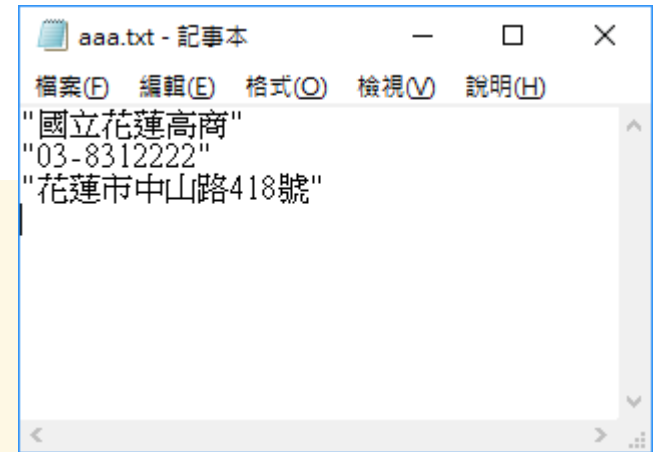
`FileClose(fp)`

- ▶ 檔案存取完畢應確實關閉檔案，釋放相關資源

檔案存取練習

► 寫入檔案

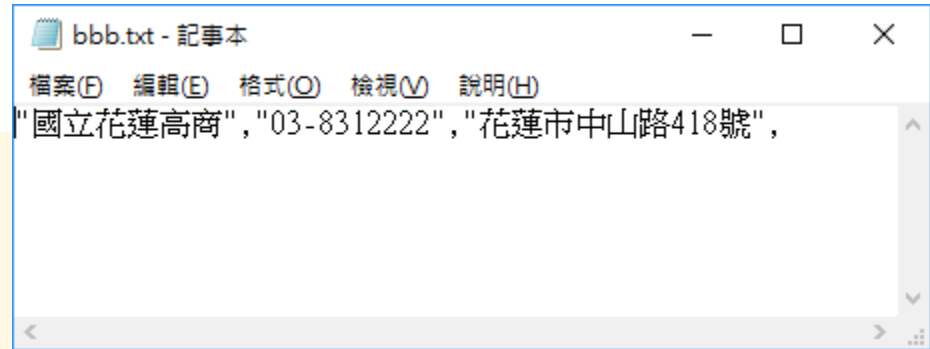
```
Dim fname As String
Dim fp As Integer
fname = "d:\aaa.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Output)
WriteLine(fp, "國立花蓮高商")
WriteLine(fp, "03-8312222")
WriteLine(fp, "花蓮市中山路418號")
FileClose(fp)
```



檔案存取練習

► 寫入檔案

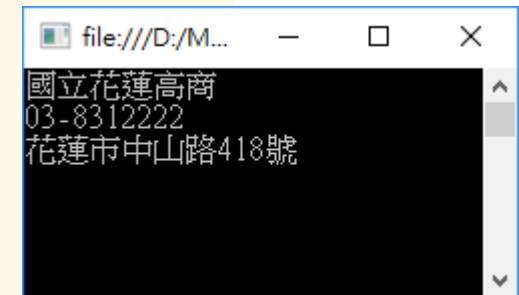
```
Dim fname As String
Dim fp As Integer
fname = "d:\bbb.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Output)
Write(fp, "國立花蓮高商")
Write(fp, "03-8312222")
Write(fp, "花蓮市中山路418號")
FileClose(fp)
```



檔案存取練習

► 讀取檔案

```
Dim fname, x As String
Dim fp As Integer
fname = "d:\bbb.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Input)
For i = 1 To 3
    Input(fp, x)
    Console.WriteLine(x)
Next
FileClose(fp)
Console.Read()
```



檔案存取練習

- ▶ 如果讀取檔案時不知道有多少資料，就不能使用FOR迴圈了
- ▶ 可以使用EOF()函數(End Of File)，判斷是否已到達檔案的最後

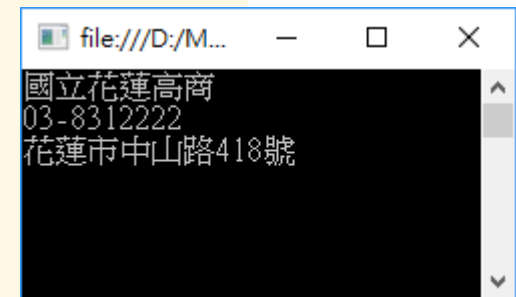
EOF (檔案指標)

- ▶ 當到達檔案結尾時傳回True，否則傳回False

檔案存取練習

- ▶ 讀取檔案，使用EOF()函數

```
Dim fname, x As String
Dim fp As Integer
fname = "d:\bbb.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Input)
Do While Not EOF(fp)
    Input(fp, x)
    Console.WriteLine(x)
Loop
FileClose(fp)
Console.Read()
```



常用File類別方法

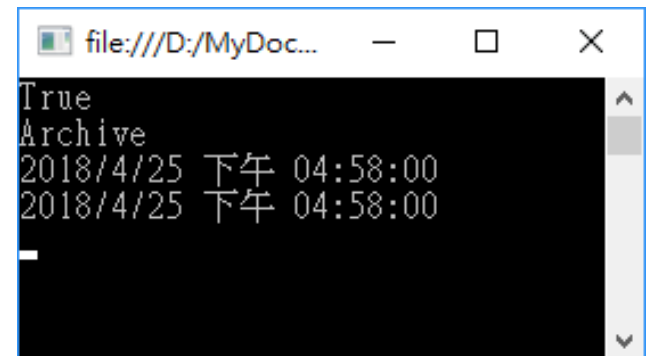
- ▶ 如果要讀取的檔案不存在怎麼辦？如果要寫入的檔案已經存在，不想覆蓋它？
- ▶ 使用File類別提供的一些方法來取得檔案資訊

模 式	說 明
<code>File.Exists(fname)</code>	判斷檔案是否存在。
<code>File.GetAttributes(fname)</code>	取得檔案屬性。
<code>File.GetCreationTime(fname)</code>	取得檔案建立時間
<code>File.GetLastAccessTime(fname)</code>	取得檔案最後修改時間

常用File類別方法

► 取得檔案資訊

```
Dim fname As String
fname = "d:\aaa.txt"
Console.WriteLine(File.Exists(fname))
Console.WriteLine(File.GetAttributes(fname).ToString)
Console.WriteLine(File.GetCreationTime(fname))
Console.WriteLine(File.GetLastAccessTime(fname))
Console.Read()
```



A screenshot of a Windows command prompt window. The title bar shows the file path "file:///D:/MyDoc...". The window contains the following output:

```
True
Archive
2018/4/25 下午 04:58:00
2018/4/25 下午 04:58:00
_
```

檔案存取練習

- ▶ 有一檔案如下，每一列是學生姓名及成績，請將其讀入，將成績由大到小排序後輸出至另一檔案
- ▶ 假設檔案在" D:\score.txt"
- ▶ 請輸出到" D:\score1.txt"

```
"陳冠軍",45  
"李雅昱",65  
"蔡翰宜",94  
"張曼鈺",78  
"莊羽韜",62  
"李琦筱",99  
"謝恩濤",58  
"紀亭玉",75  
"黃子宏",86  
"杜瑾媛",100
```

檔案存取練習

▶ 主程式

```
Dim name(1), fname, x As String
Dim score(1), fp, m As Integer
```

宣告成
全域變數

```
Sub Main()
    ReadTxt()      '讀入檔案
    PrintArr()     '印出原資料
    ArrSort()      '排序
    PrintArr()     '印出排序後資料
    WriteTxt()     '寫入檔案
    Console.Read() '暫停畫面
End Sub
```

將每一個動
作都寫成副
程式

檔案存取練習

► 讀入檔案資料

```
Sub ReadTxt()  
    fname = "d:\score.txt"  
    fp = FreeFile()  
    FileOpen(fp, fname, OpenMode.Input)  
    Do While Not EOF(fp)  
        Input(fp, x)  
        name(UBound(name)) = x  
        Input(fp, m)  
        score(UBound(score)) = m  
        ReDim Preserve name(UBound(name) + 1)  
        ReDim Preserve score(UBound(score) + 1)  
    Loop  
    FileClose(fp)  
    Console.WriteLine("讀入 " & fname & " 完畢")  
End Sub
```

不確定有多少
資料，故使用
動態陣列

檔案存取練習

▶ 印出陣列內容

```
Sub PrintArr()  
    For i = 1 To UBound(name) - 1  
        Console.WriteLine(name(i) & ", " & score(i))  
    Next  
    Console.WriteLine()  
End Sub
```

檔案存取練習

► 由大到小排序

```
Sub ArrSort()  
    Dim tmpm As Integer  
    Dim tmps As String  
    For i = 1 To UBound(score) - 2      '使用選擇排序  
        For j = i + 1 To UBound(score) - 1  
            If score(i) < score(j) Then  
                tmpm = score(i)          '交換分數  
                score(i) = score(j)  
                score(j) = tmpm  
                tmps = name(i)           '交換姓名  
                name(i) = name(j)  
                name(j) = tmps  
            End If  
        Next  
    Next  
End Sub
```

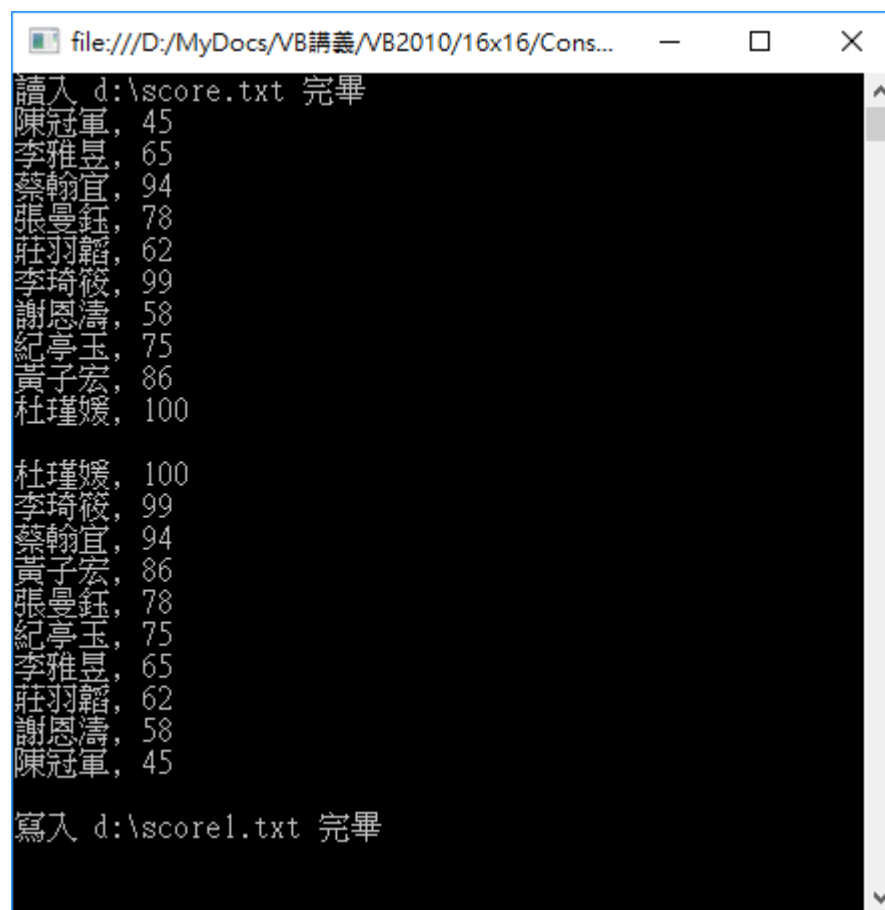
檔案存取練習

▶ 寫入檔案

```
Sub WriteTxt()  
    fname = "d:\score1.txt"  
    fp = FreeFile()  
    FileOpen(fp, fname, OpenMode.Output)  
    For i = 1 To UBound(score) - 1  
        Write(fp, name(i))  
        WriteLine(fp, score(i))  
    Next  
    FileClose(fp)  
    Console.WriteLine("寫入 " & fname & " 完畢")  
End Sub
```

檔案存取練習

▶ 執行結果



```
file:///D:/MyDocs/VB講義/VB2010/16x16/Cons...  —  □  ×
讀入 d:\score.txt 完畢
陳冠軍, 45
李雅昱, 65
蔡翰宜, 94
張曼鈺, 78
莊羽韜, 62
李琦筱, 99
謝恩濤, 58
紀亭玉, 75
黃子宏, 86
杜瑾媛, 100
杜瑾媛, 100
李琦筱, 99
蔡翰宜, 94
黃子宏, 86
張曼鈺, 78
紀亭玉, 75
李雅昱, 65
莊羽韜, 62
謝恩濤, 58
陳冠軍, 45
寫入 d:\score1.txt 完畢
```

完整程式

Module Module1

```
Dim name(1), fname, x As String
```

```
Dim score(1), fp, m As Integer
```

Sub Main()

```
ReadTxt() '讀入檔案
```

```
PrintArr() '印出原資料
```

```
ArrSort() '排序
```

```
PrintArr() '印出排序後資料
```

```
WriteTxt() '寫入檔案
```

```
Console.Read()
```

```
End Sub
```

Sub ReadTxt()

```
fname = "d:\score.txt"
```

```
fp = FreeFile()
```

```
FileOpen(fp, fname, OpenMode.Input)
```

```
Do While Not EOF(fp)
```

```
    Input(fp, x)
```

```
    name(UBound(name)) = x
```

```
    Input(fp, m)
```

```
    score(UBound(score)) = m
```

```
    ReDim Preserve name(UBound(name) + 1)
```

```
    ReDim Preserve score(UBound(score) + 1)
```

```
Loop
```

```
FileClose(fp)
```

```
Console.WriteLine("讀入 " & fname & " 完畢")
```

```
End Sub
```

Sub PrintArr()

```
For i = 1 To UBound(name) - 1
```

```
    Console.WriteLine(name(i) & ", " & score(i))
```

```
Next
```

```
Console.WriteLine()
```

```
End Sub
```

Sub ArrSort()

```
Dim tmpm As Integer
```

```
Dim tmps As String
```

```
For i = 1 To UBound(score) - 2 '選擇排序
```

```
    For j = i + 1 To UBound(score) - 1
```

```
        If score(i) < score(j) Then
```

```
            tmpm = score(i) '交換分數
```

```
            score(i) = score(j)
```

```
            score(j) = tmpm
```

```
            tmps = name(i) '交換姓名
```

```
            name(i) = name(j)
```

```
            name(j) = tmps
```

```
        End If
```

```
    Next
```

```
Next
```

```
End Sub
```

Sub WriteTxt()

```
fname = "d:\score1.txt"
```

```
fp = FreeFile()
```

```
FileOpen(fp, fname, OpenMode.Output)
```

```
For i = 1 To UBound(score) - 1
```

```
    Write(fp, name(i))
```

```
    WriteLine(fp, score(i))
```

```
Next
```

```
FileClose(fp)
```

```
Console.WriteLine("寫入 " & fname & " 完畢")
```

```
End Sub
```

```
End Module
```

綜合練習



綜合練習一、身分證字號檢查

▶ 說明：

- ▶ 身分證字號的每個字元代表的意義如下,例如：A123456789
- ▶ (1)第一個字元代表地區，轉換方式為：A轉換成1,0兩個字元，B轉換成1,1，C轉換成1,2

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
10	11	12	13	14	15	16	17	34	18	19	20	21	22	35	23	24	25	26	27	28	29	32	30	31	33

- ▶ (2)第二個字元代表性別，1代表男性，2代表女性
- ▶ (3)第三個字元到第九個字元為流水號碼。
- ▶ (4)第十個字元為檢查號碼。

綜合練習一、身分證字號檢查

- 檢查碼產生的規則為：

每個相對應數字乘上權數										檢查碼
A		1	2	3	4	5	6	7	8	9
1	0	x8	x7	x6	x5	x4	x3	x2	x1	
x1	X9									

- 將乘上權數後之積相加

$$\begin{array}{rclcl}
 A \left\{ \begin{array}{l} 1 \times 1 = 1 \\ 0 \times 9 = 0 \\ 1 \times 8 = 8 \\ 2 \times 7 = 14 \\ 3 \times 6 = 18 \\ 4 \times 5 = 20 \\ 5 \times 4 = 20 \\ 6 \times 3 = 18 \\ 7 \times 2 = 14 \\ +) 8 \times 1 = 8 \end{array} \right. \\
 \hline
 \qquad \qquad \qquad 121
 \end{array}$$

綜合練習一、身分證字號檢查

- ▶ 相加後之值除以模數10取其餘數

$$\text{Ex : } 1\ 2\ 1 / 10 = 12 \dots \text{餘數 } 1$$

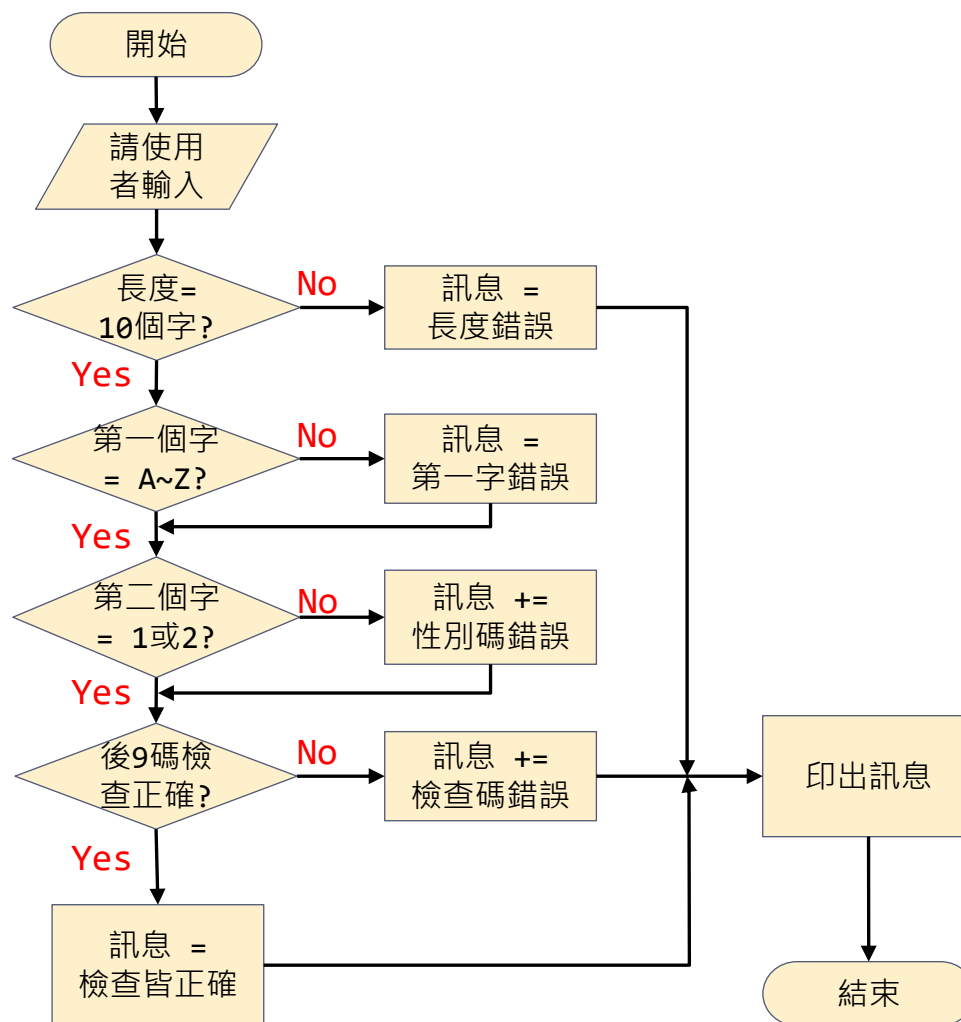
- ▶ 由模數減去餘數得檢查碼

$$\text{Ex : } 10 - 1 = 9$$

- ▶ 若餘數為0時，則設定其檢查碼為0

綜合練習一、身分證字號檢查

► 流程圖：



綜合練習一、身分證字號檢查

▶ 參考程式：

```
Sub Main()
    Dim ID, ErrMsg As String
    Dim alpha(26) As String
    Dim N, N0, N1, Sum As Integer
    ErrMsg = ""
    '建立字母對照表
    alpha(1) = "10" : alpha(14) = "22"
    alpha(2) = "11" : alpha(15) = "35"
    alpha(3) = "12" : alpha(16) = "23"
    alpha(4) = "13" : alpha(17) = "24"
    alpha(5) = "14" : alpha(18) = "25"
    alpha(6) = "15" : alpha(19) = "26"
    alpha(7) = "16" : alpha(20) = "27"
    alpha(8) = "17" : alpha(21) = "28"
    alpha(9) = "34" : alpha(22) = "29"
    alpha(10) = "18" : alpha(23) = "32"
    alpha(11) = "19" : alpha(24) = "30"
    alpha(12) = "20" : alpha(25) = "31"
    alpha(13) = "21" : alpha(26) = "33"

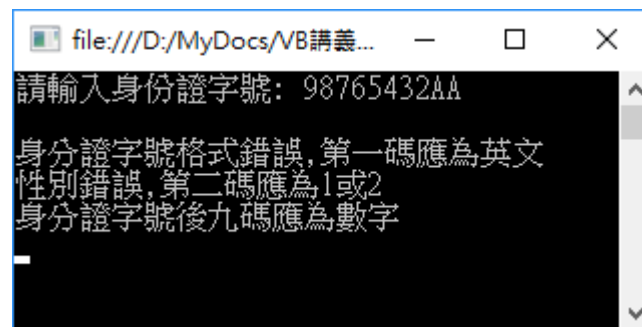
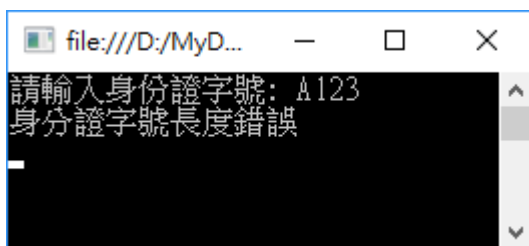
    Console.Write("請輸入身分證字號: ")
    ID = Console.ReadLine().ToUpper
    If Len(ID) <> 10 Then '檢查長度是否為10個字
        ErrMsg = "身分證字號長度錯誤"
    Else '檢查第一碼是否為英文字
        If Mid(ID, 1, 1) < "A" Or Mid(ID, 1, 1) > "Z" Then
            ErrMsg = ErrMsg & vbCrLf & "身分證字號格式錯誤, 第一碼應為英文"
        End If
        If Mid(ID, 2, 1) <> "1" And Mid(ID, 2, 1) <> "2" Then '檢查性別
            ErrMsg = ErrMsg & vbCrLf & "性別錯誤, 第二碼應為1或2"
        End If
        For i = 2 To 10 '檢查後九碼是否為數字
            If Mid(ID, i, 1) < "0" Or Mid(ID, i, 1) > "9" Then
                ErrMsg = ErrMsg & vbCrLf & "身分證字號後九碼應為數字"
                Exit For
            End If
        Next
    End If
End Sub
```

```
If ErrMsg = "" Then '計算檢查碼是否正確
    N0 = Val(Mid(alpha(Asc(Mid(ID, 1, 1))-64), 1, 1))
    N1 = Val(Mid(alpha(Asc(Mid(ID, 1, 1))-64), 2, 1))*9
    N = 0
    For i = 2 To 9
        N = N + Mid(ID, i, 1) * (10 - i)
    Next
    Sum = (N0 + N1 + N) Mod 10
    If Sum <> 0 Then Sum = 10 - Sum
    If Sum <> Val(Mid(ID, 10, 1)) Then
        ErrMsg = ErrMsg & vbCrLf & "身分證字號檢查碼不符"
    End If
End If
End If
'印出訊息
If ErrMsg = "" Then
    Console.WriteLine("這是正確的身分證字號")
Else
    Console.WriteLine(ErrMsg)
End If
Console.Read()
End Sub
```

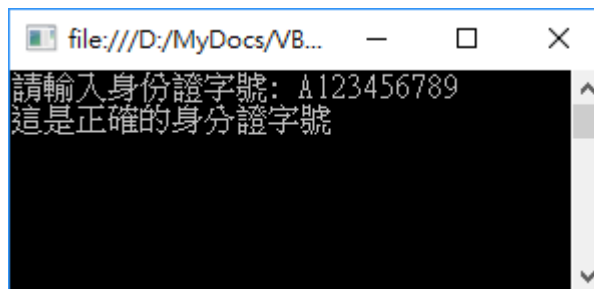
綜合練習一、身分證字號檢查

▶ 執行結果：

▶ 1.輸入錯誤號碼：



▶ 2.輸入正確號碼：



綜合練習一、身分證字號檢查

▶ 程式中用到的函數：

函數	用法	說明
LEN()	LEN(ID)	傳回字串或字串變數的字元數(或稱長度)
MID()	MID(ID,m,n)	傳回字串或字串變數第m個位置起共n個
ASC()	ASC(x)	傳回字元的ASCII碼(十進制)
VAL()	VAL(x)	將文字轉成數值

▶ 程式中用到的方法：

方法	說明
String.ToUpper	將英文字母轉換成大寫
String.ToLower	將英文字母轉換成小寫

綜合練習一、身分證字號檢查

▶ ASCII表(部分)

Name	Hex	Dec
. (period)	2E	046
0	30	048
1	31	049
2	32	050
3	33	051
4	34	052
5	35	053
6	36	054
7	37	055
8	38	056
9	39	057

Name	Hex	Dec
A	41	065
B	42	066
C	43	067
D	44	068
E	45	069
F	46	070
G	47	071
H	48	072
I	49	073
J	4A	074
K	4B	075

Name	Hex	Dec
L	4C	076
M	4D	077
N	4E	078
O	4F	079
P	50	080
Q	51	081
R	52	082
S	53	083
T	54	084
U	55	085
V	56	086

Name	Hex	Dec
W	57	087
X	58	088
Y	59	089
Z	5A	090

綜合練習一、身分證字號檢查

▶ 分段說明：

```
Dim ID, ErrMsg As String
Dim alpha(26) As String
Dim N, N0, N1, Sum As Integer
ErrMsg = ""
'建立字母對照表
alpha(1) = "10" : alpha(14) = "22"
alpha(2) = "11" : alpha(15) = "35"
alpha(3) = "12" : alpha(16) = "23"
alpha(4) = "13" : alpha(17) = "24"
alpha(5) = "14" : alpha(18) = "25"
alpha(6) = "15" : alpha(19) = "26"
alpha(7) = "16" : alpha(20) = "27"
alpha(8) = "17" : alpha(21) = "28"
alpha(9) = "34" : alpha(22) = "29"
alpha(10) = "18" : alpha(23) = "32"
alpha(11) = "19" : alpha(24) = "30"
alpha(12) = "20" : alpha(25) = "31"
alpha(13) = "21" : alpha(26) = "33"
```

變數：

ID:使用者輸入的身分證字號

ErrMsg:存放錯誤訊息供輸出

依規定字母需轉換成兩數字碼，
先產生陣列供查詢

取得字母的ASCII值(65~90)，
然後減掉64，即可對應到
alpha陣列

綜合練習一、身分證字號檢查

```
Console.Write("請輸入身分證字號： ")
ID = Console.ReadLine().ToUpper
If Len(ID) <> 10 Then '檢查長度是否為10個字
    ErrMsg = "身分證字號長度錯誤"
Else '檢查第一碼是否為英文字
    If Mid(ID, 1, 1) < "A" Or Mid(ID, 1, 1) > "Z" Then
        ErrMsg = ErrMsg & vbCrLf & "身分證字號格式錯誤,第一碼應為英文"
    End If
    If Mid(ID, 2, 1) <> "1" And Mid(ID, 2, 1) <> "2" Then '檢查性別
        ErrMsg = ErrMsg & vbCrLf & "性別錯誤,第二碼應為1或2"
    End If
    For i = 2 To 10 '檢查後九碼是否為數字
        If Mid(ID, i, 1) < "0" Or Mid(ID, i, 1) > "9" Then
            ErrMsg = ErrMsg & vbCrLf & "身分證字號後九碼應為數字"
            Exit For
        End If
    Next
```

綜合練習一、身分證字號檢查

```
If ErrMsg = "" Then '計算檢查碼是否正確
    N0 = Val(Mid(alpha(Asc(Mid(ID, 1, 1))-64), 1, 1))
    N1 = Val(Mid(alpha(Asc(Mid(ID, 1, 1))-64), 2, 1))*9
    N = 0
    For i = 2 To 9
        N = N + Mid(ID, i, 1) * (10 - i)
    Next
    Sum = (N0 + N1 + N) Mod 10
    If Sum <> 0 Then Sum = 10 - Sum
    If Sum <> Val(Mid(ID, 10, 1)) Then
        ErrMsg = ErrMsg & vbCrLf & "身分證字號檢查碼不符"
    End If
End If
End If
```

綜合練習一、身分證字號檢查

▶ '印出訊息

```
If ErrMsg = "" Then  
    Console.WriteLine("這是正確的身分證字號")  
Else  
    Console.WriteLine(ErrMsg)  
End If  
Console.Read()
```



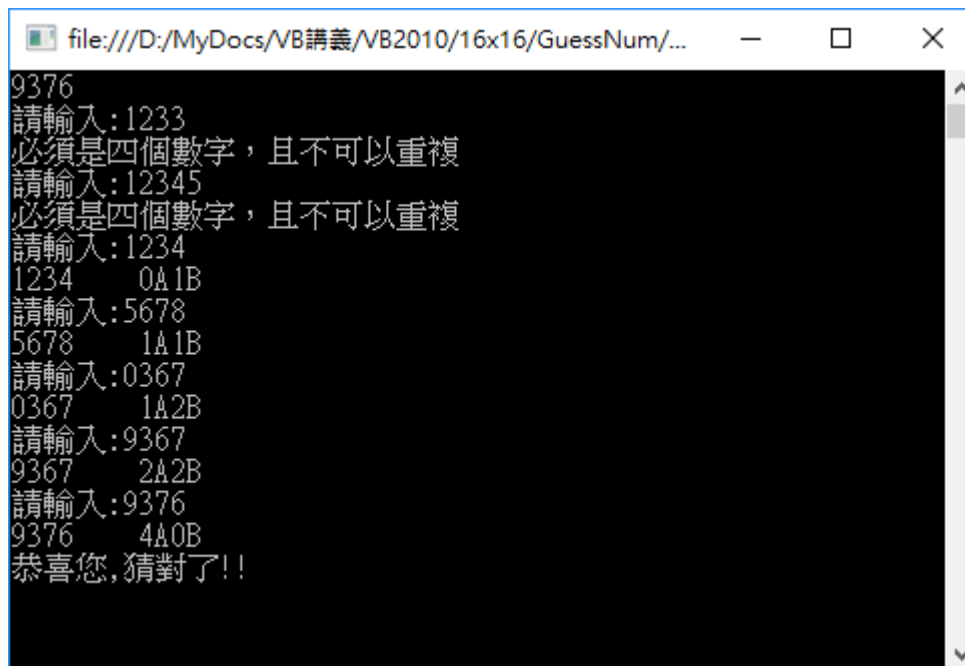
綜合練習二、猜數字遊戲

- ▶ 寫一猜數字程式,電腦隨機決定一個四位數字(每個數字不可重複),由使用者猜,若數值對,但位置不對,給一個B,若數值對,位置也對,給一個A,直到4A為止(即全部答對)
- ▶ 例 :

例：答案為2478	
輸入	回答
1234	0A2B
5678	2A0B
2745	1A2B
2478	答對了

綜合練習二、猜數字遊戲

▶ 執行結果：



```
file:///D:/MyDocs/VB講義/VB2010/16x16/GuessNum/...
9376
請輸入:1233
必須是四個數字，且不可以重複
請輸入:12345
必須是四個數字，且不可以重複
請輸入:1234
1234    0A1B
請輸入:5678
5678    1A1B
請輸入:0367
0367    1A2B
請輸入:9367
9367    2A2B
請輸入:9376
9376    4A0B
恭喜您,猜對了!!
```

綜合練習二、猜數字遊戲

▶ 參考寫法：

```
U 猜數字
Module Module1
    Dim a1, a2, a3, a4 As Integer '使用者輸入的猜測
    Dim q1, q2, q3, q4 As Integer '電腦答案

    0 個參考
    Sub Main()
        Randomize()
        Dim X As String
        '產生四個, 互不相同的數字
        q1 = Int(10 * Rnd())
        Do
            q2 = Int(10 * Rnd())
            Loop While (q1 = q2)
        Do
            q3 = Int(10 * Rnd())
            Loop While (q3 = q1 Or q3 = q2)
        Do
            q4 = Int(10 * Rnd())
            Loop While (q4 = q1 Or q4 = q2 Or q4 = q3)
        Console.WriteLine(q1 & q2 & q3 & q4) '顯示答案, 測試程式時使用
        '使用者輸入直到猜對
        Do
            Console.Write("請輸入:")
            X = Trim(Console.ReadLine())
            Loop Until CheckOK(X)
            Console.WriteLine("恭喜您, 猜對了!!")
            Console.ReadLine()
        End Sub

    1 個參考
    Function CheckOK(ByVal x As String)
        Dim A, B As Integer
        A = 0 : B = 0
        a1 = Val(Mid(x, 1, 1))
        a2 = Val(Mid(x, 2, 1))
        a3 = Val(Mid(x, 3, 1))
        a4 = Val(Mid(x, 4, 1))
        If (a1 = a2 Or a1 = a3 Or a1 = a4 Or a2 = a3 Or a2 = a4 Or a3 = a4 Or Len(x) <> 4 Then
            Console.WriteLine("必須是四位數字, 且不可以重複")
        Else
            Console.Write(x & vbCrLf) '先印出使用者的輸入
            If a1 = q1 Then '檢查第一位數字
                A = A + 1
            Else
                If (a1 = q2 Or a1 = q3 Or a1 = q4) Then B = B + 1
            End If
            If a2 = q2 Then '檢查第二位數字
                A = A + 1
            Else
                If (a2 = q1 Or a2 = q3 Or a2 = q4) Then B = B + 1
            End If
            If a3 = q3 Then '檢查第三位數字
                A = A + 1
            Else
                If (a3 = q1 Or a3 = q2 Or a3 = q4) Then B = B + 1
            End If
            If a4 = q4 Then '檢查第四位數字
                A = A + 1
            Else
                If (a4 = q1 Or a4 = q2 Or a4 = q3) Then B = B + 1
            End If
            '印出檢查結果及回傳
            Console.WriteLine(A & "A" & B & "B")
            If A = 4 Then Return True Else Return False
        End If
        Return False
    End Function
End Module
```

綜合練習二、猜數字遊戲

▶ 分段說明：

```
Dim a1, a2, a3, a4 As Integer '使用者輸入的猜測
Dim q1, q2, q3, q4 As Integer '存放答案

Sub Main()
    Randomize()
    Dim X As String
    '產生題目,四個不重複的數字
    q1 = Int(10 * Rnd())
    Do
        q2 = Int(10 * Rnd())
    Loop While (q1 = q2)
    Do
        q3 = Int(10 * Rnd())
    Loop While (q3 = q1 Or q3 = q2)
    Do
        q4 = Int(10 * Rnd())
    Loop While (q4 = q1 Or q4 = q2 Or q4 = q3)
    Console.WriteLine(q1 & q2 & q3 & q4) '顯示答案,測試程式時使用
```

綜合練習二、猜數字遊戲

► 分段說明：

```
'請使用者輸入直到答對
Do
    Console.Write("請輸入:")
    X = Trim(Console.ReadLine())
Loop Until CheckOK(X) '呼叫函數
Console.WriteLine("恭喜您,猜對了!!")
Console.ReadLine()
End Sub
```

綜合練習二、猜數字遊戲

► 分段說明：

```
Function CheckOK(ByVal x As String)
    Dim A, B As Integer
    A = 0 : B = 0
    a1 = Val(Mid(x, 1, 1))
    a2 = Val(Mid(x, 2, 1))
    a3 = Val(Mid(x, 3, 1))
    a4 = Val(Mid(x, 4, 1))
    If (a1 = a2 Or a1 = a3 Or a1 = a4 Or a2 = a3 Or a2 = a4 Or a3 = a4)
        Or Len(x) <> 4 Then
        Console.WriteLine("必須是四個數字，且不可以重複")
    Else
        Console.Write(x & vbTab)    '先印出使用者的輸入
```

綜合練習二、猜數字遊戲

▶ 分段說明：

```
If a1 = q1 Then '檢查第一個數字
    A = A + 1
Else
    If (a1 = q2 Or a1 = q3 Or a1 = q4) Then B = B + 1
End If
If a2 = q2 Then '檢查第二個數字
    A = A + 1
Else
    If (a2 = q1 Or a2 = q3 Or a2 = q4) Then B = B + 1
End If
If a3 = q3 Then '檢查第三個數字
    A = A + 1
Else
    If (a3 = q2 Or a3 = q1 Or a3 = q4) Then B = B + 1
End If
If a4 = q4 Then '檢查第四個數字
    A = A + 1
Else
    If (a4 = q2 Or a4 = q3 Or a4 = q1) Then B = B + 1
End If
```

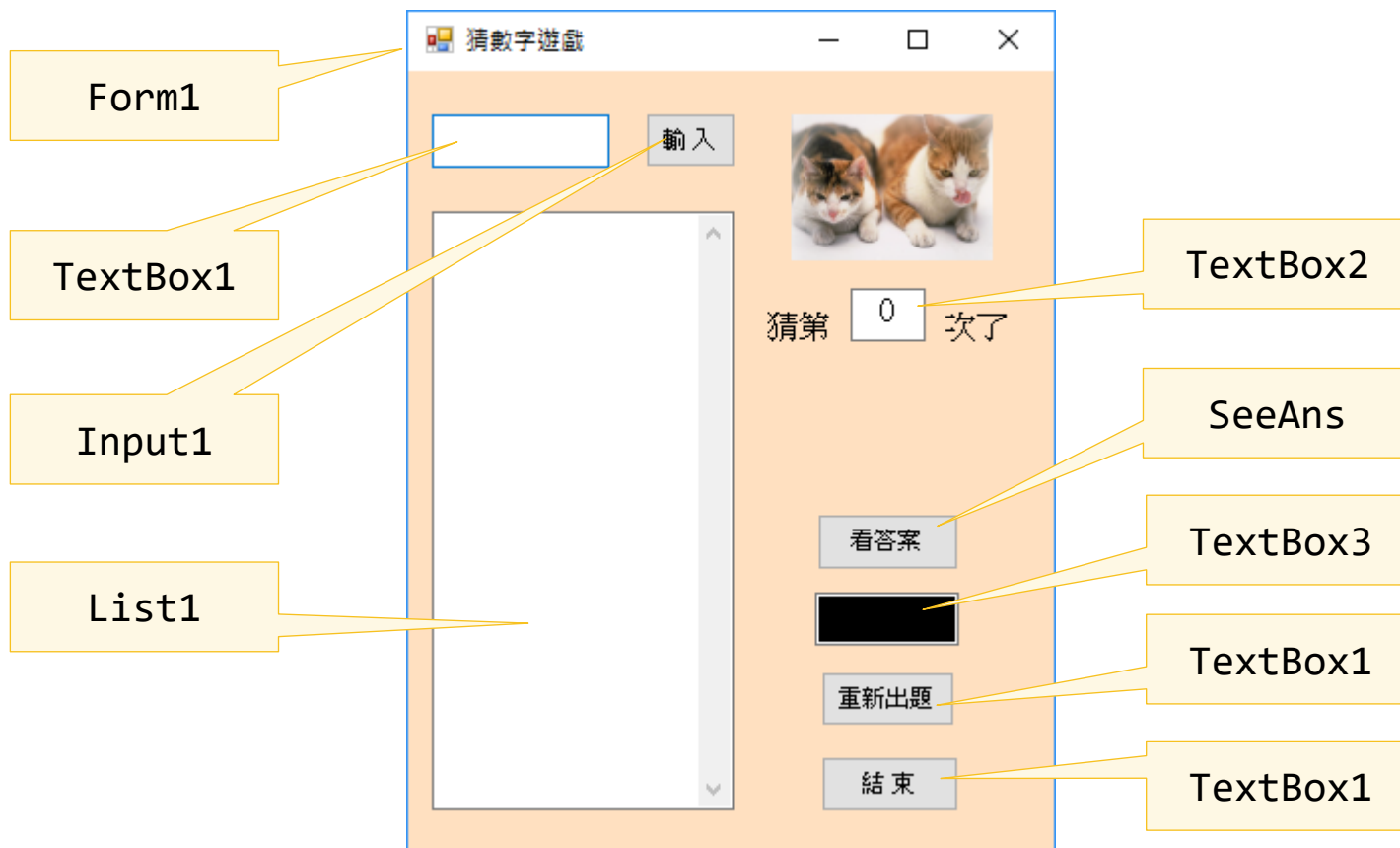
綜合練習二、猜數字遊戲

▶ 分段說明：

```
' 印出檢查結果及回傳
Console.WriteLine(A & "A" & B & "B")
If A = 4 Then
    Return True
Else
    Return False
End If
End If
Return False
End Function
```

綜合練習三、猜數字遊戲視窗版

► 將前一遊戲改成視窗版：



▶ 參考程式：(請參閱講義)

▶ 155

綜合練習三、猜數字遊戲視窗版

► 執行結果：

