

程式語言(programming language)

使用Visual Basic 劉和師

Part 2

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

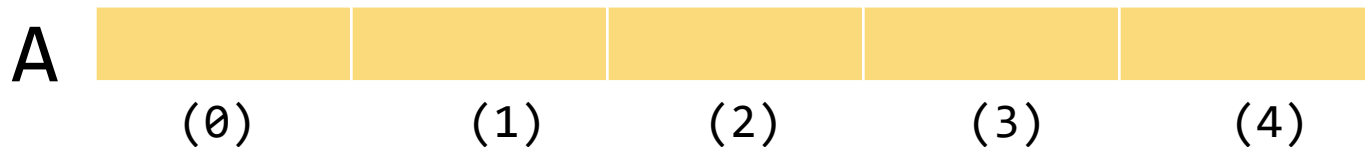
- ▶ 宣告：任何陣列在使用前必須先宣告,格式為

Dim 陣列名稱(最大註標值) **As** 資料型態

- ▶ Ex :

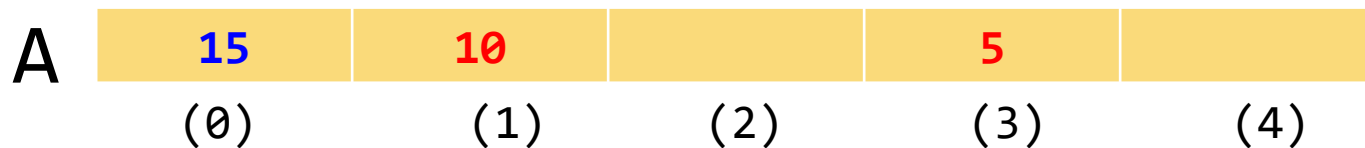
Dim A(4) As Integer

- ▶ 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



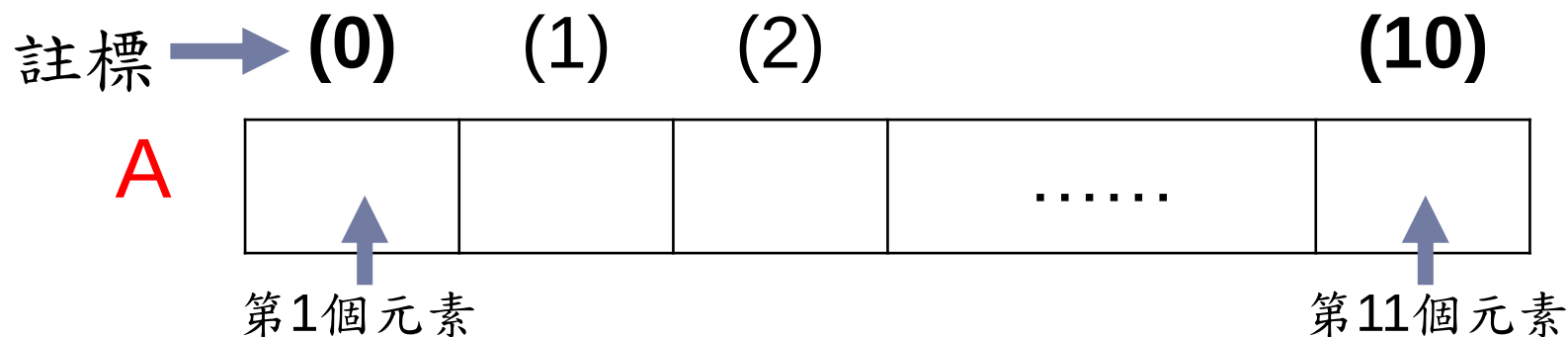
一維陣列

- ▶ 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

- 下圖為一維陣列的概念，陣列名稱為A，依陣列的編號(n)來存取，例如A(10)：



注意：陣列註標為正整數,而且是從0開始

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

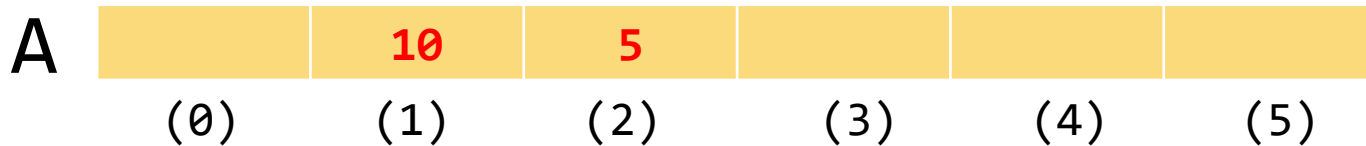
A(0), A(1), A(2), A(3), A(4), A(5)

- ▶ 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 :

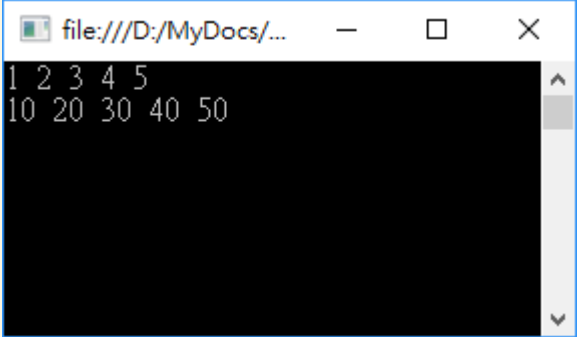
```
For I = 1 to 10  
    A(I) = A(I) + 1  
Next
```

- ▶ 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

- ▶ 1. 利用一維陣列求10個數字的計算平均值。
- ▶ 2. 輸入10個數字至陣列A，再輸入x，判斷x是否存在於A陣列中，如果存在，輸出所在的註標(索引)值，若不存在，則告知不存在。
- ▶ 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

一維陣列練習

▶ 1.參考程式：

```
Sub Main()  
    Dim A(10) As Integer    '宣告一維陣列A(10)  
    Dim x As Single  
    For i = 1 To 10  
        Console.WriteLine("Please input the value: ")  
        A(i) = Console.ReadLine() '將輸入的數值存入陣列  
    Next  
    x = 0  
    For i = 1 To 10 '依序加總陣列元素  
        x += A(i)  
    Next  
    Console.WriteLine("x = " & x / 10)  
    Console.Read()  
End Sub
```

- ▶ 雖然陣列宣告後其所有元素初值為0，但任何變數使用前最好還是要確定其內含值

一維陣列練習

▶ 2.參考程式：

```
Sub Main()  
    Dim A(10), k, x As Integer    '宣告一維陣列A(10)  
    For i = 1 To 10  
        Console.Write("Please input the value: ")  
        A(i) = Console.ReadLine() '將輸入的數值存入陣列  
    Next  
    Console.Write("Please input the value youy want to search: ")  
    x = Console.ReadLine()  
    k = 0    '如果最後k仍然是0表示沒找到  
    For i = 1 To 10  
        If x = A(i) Then    '找到，設定k並印出i值  
            k = 1  
            Console.WriteLine("index = " & i)  
        End If  
    Next  
    If k = 0 Then Console.WriteLine(x & " does not exist ")  
    Console.Read()  
End Sub
```

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} }
```

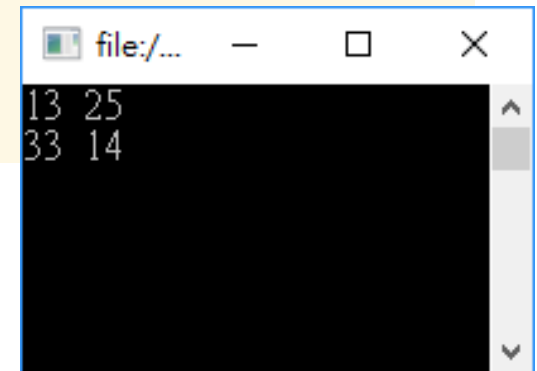
- ▶ 陣列A內容：

$A(0,0)$ 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()
```



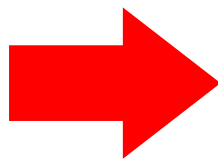
A screenshot of a Windows console window. The title bar shows 'file:/...' and standard window controls. The console output displays two lines of text: '13 25' on the first line and '33 14' on the second line. The rest of the console is black.

二維陣列練習

▶ 練習：陣列轉置(一)

1	2	3
4	5	6
7	8	9

A

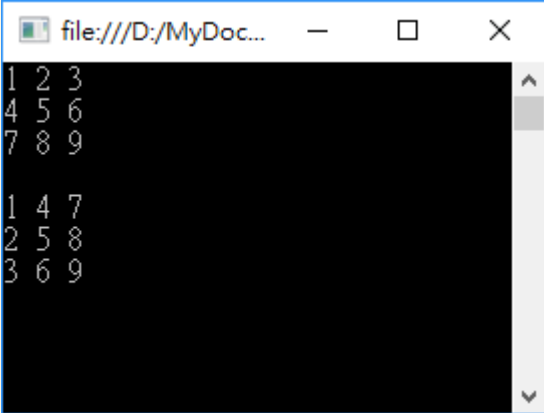


1	4	7
2	5	8
3	6	9

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()
```



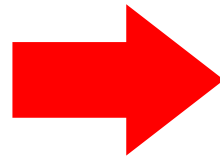
```
file:///D:/MyDoc...
1 2 3
4 5 6
7 8 9
1 4 7
2 5 8
3 6 9
```

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()
```

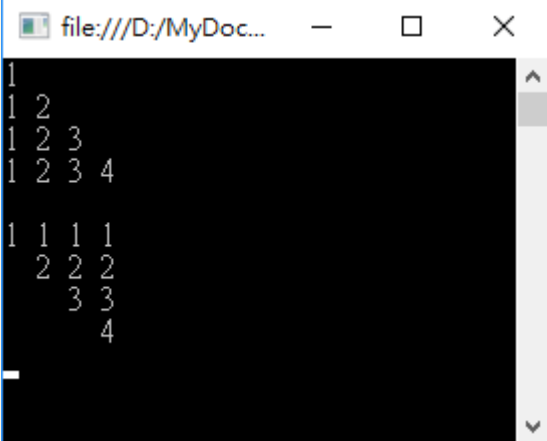
```
For i = 1 To 4 '印出陣列
    For j = 1 To 4
        If A(i, j) <> 0 Then
            Console.Write(A(i, j) & " ")
        Else
            Console.Write(" ")
        End If
    Next
    Console.WriteLine()
```

```
Next
```

```
Console.WriteLine()
```

```
Next
```

```
Console.Read()
```



```
file:///D:/MyDoc...
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

二維陣列練習

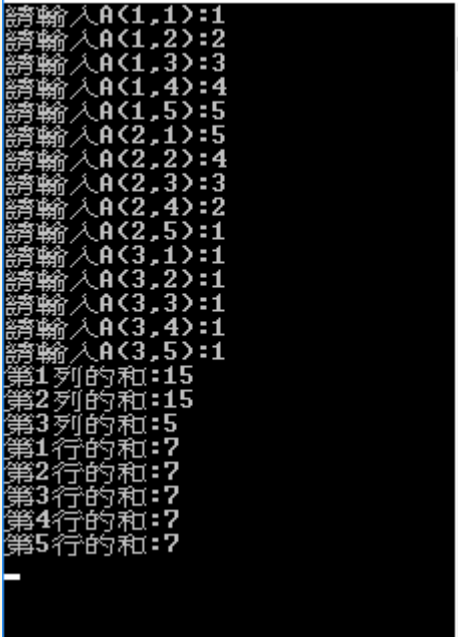
- 寫一程式，將15個數字存入3X5的二維陣列中，求每一行及每一列的數字的和。

行					
列					= ?
					= ?
					= ?
	= ?	= ?	= ?	= ?	= ?

二維陣列練習

▶ 參考寫法：

```
Dim A(3, 5), sum As Integer
Sub Main()
    For i = 1 To 3 '輸入資料
        For j = 1 To 5
            Console.Write("請輸入A(" & i & "," & j & "):")
            A(i, j) = Console.ReadLine()
        Next
    Next
    For i = 1 To 3 '計算列
        sum = 0
        For j = 1 To 5
            sum += A(i, j)
        Next
        Console.WriteLine("第" & i & "列的和:" & sum)
    Next
    For j = 1 To 5 '計算行
        sum = 0
        For i = 1 To 3
            sum += A(i, j)
        Next
        Console.WriteLine("第" & j & "行的和:" & sum)
    Next
    Console.Read()
End Sub
```



```
請輸入A(1,1):1
請輸入A(1,2):2
請輸入A(1,3):3
請輸入A(1,4):4
請輸入A(1,5):5
請輸入A(2,1):5
請輸入A(2,2):4
請輸入A(2,3):3
請輸入A(2,4):2
請輸入A(2,5):1
請輸入A(3,1):1
請輸入A(3,2):1
請輸入A(3,3):1
請輸入A(3,4):1
請輸入A(3,5):1
第1列的和:15
第2列的和:15
第3列的和:5
第1行的和:7
第2行的和:7
第3行的和:7
第4行的和:7
第5行的和:7
```

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...



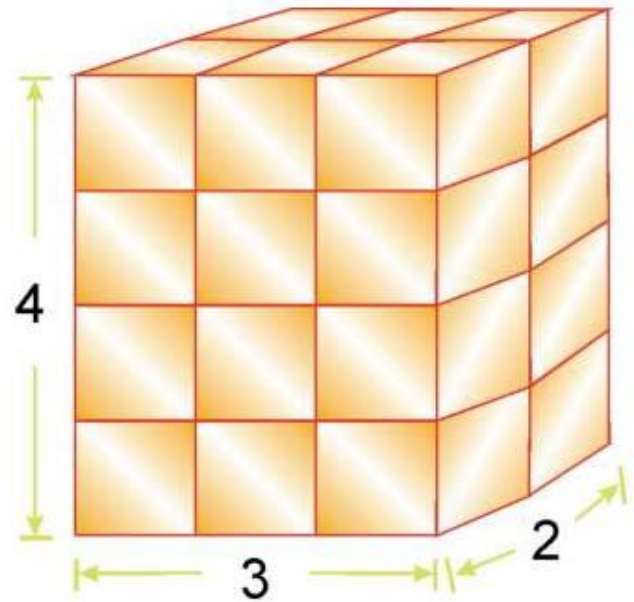
三維陣列

▶ 宣告：

Dim 陣列名稱(註標一,註標二,註標三) As 資料型態

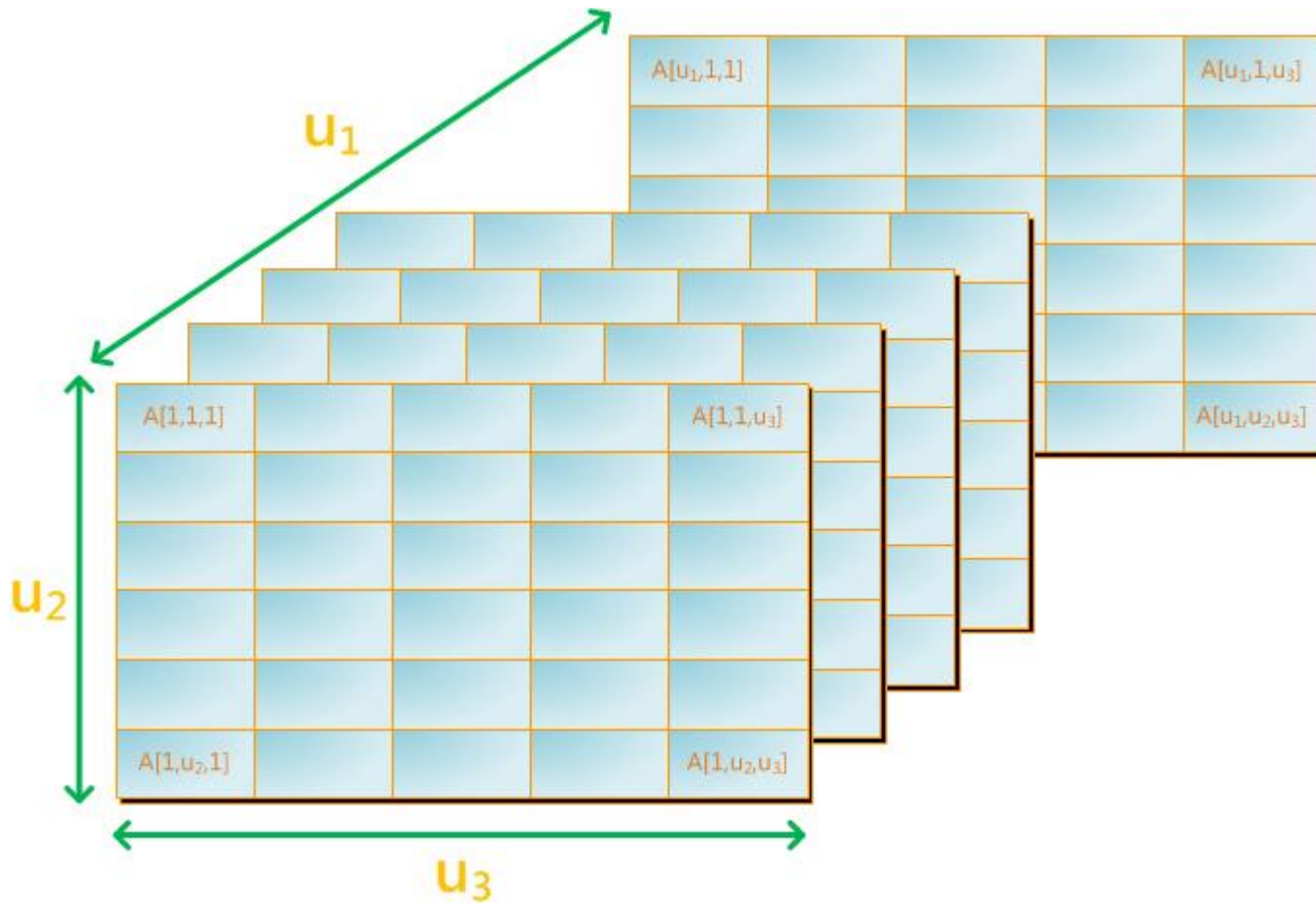
▶ 例如：

Dim A(4,3,2) As Integer



三維陣列

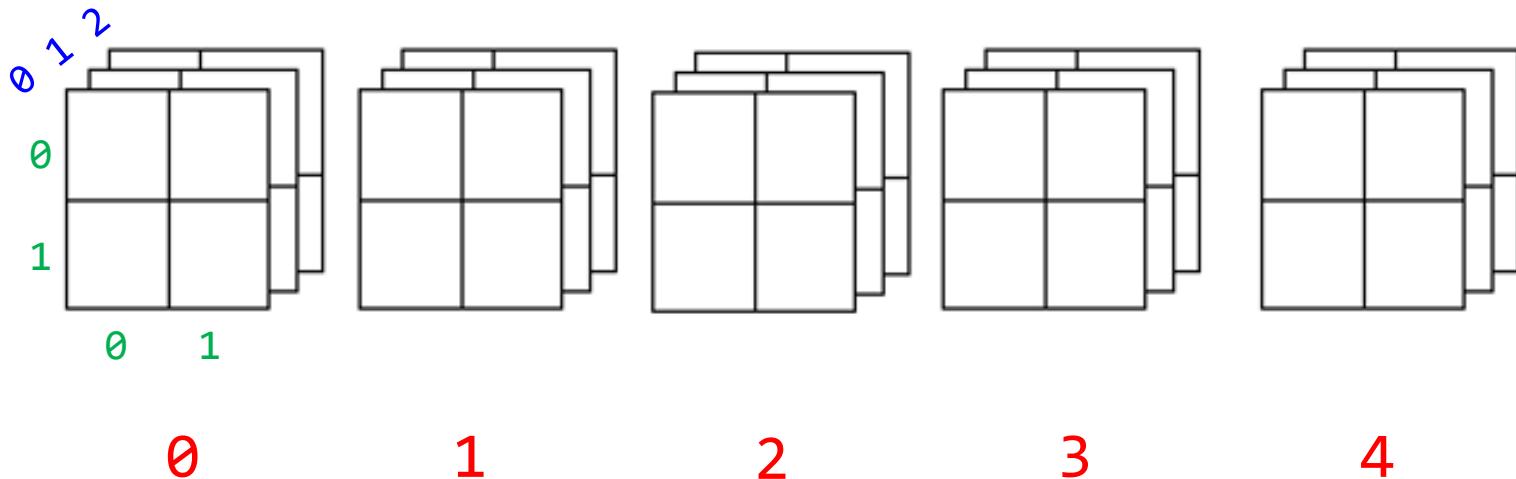
- ▶ 可想成是多個二維陣列



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

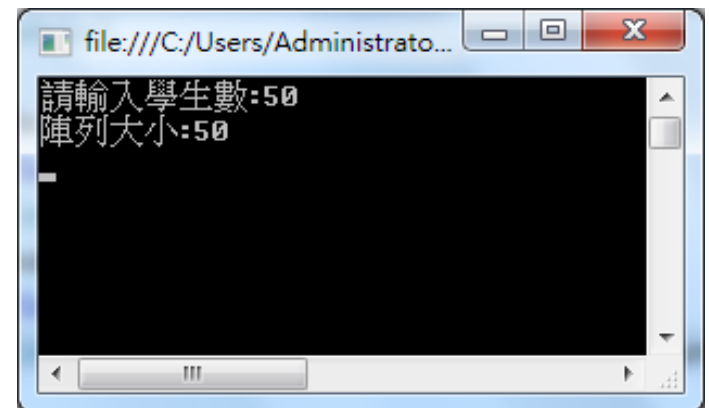
‘在程式中指定變數值

‘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)
```

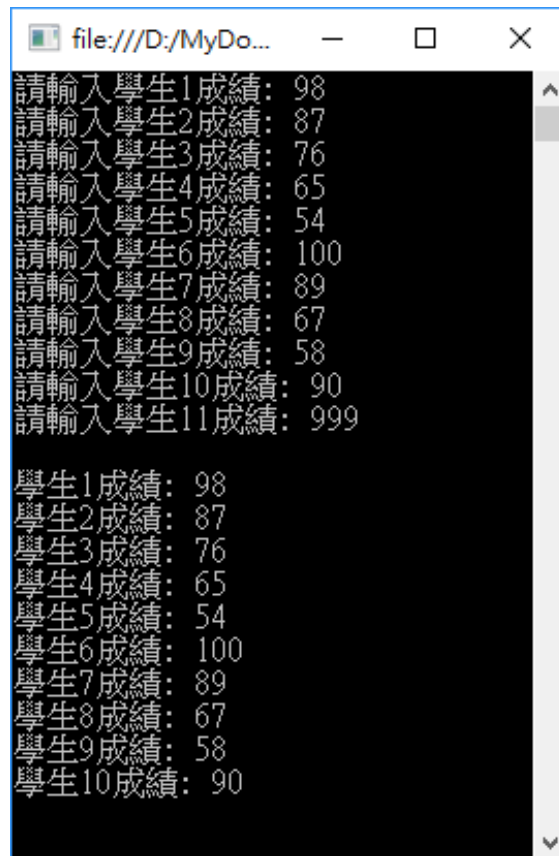
‘可以

‘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()
```

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
```

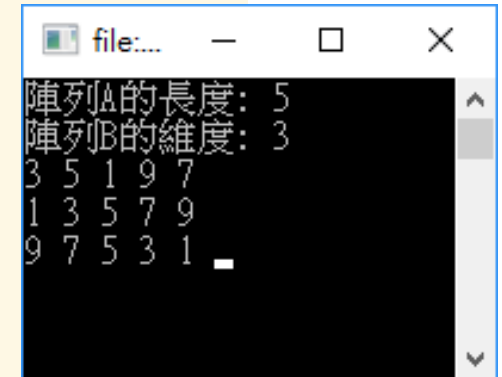
常用陣列屬性與方法

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
```

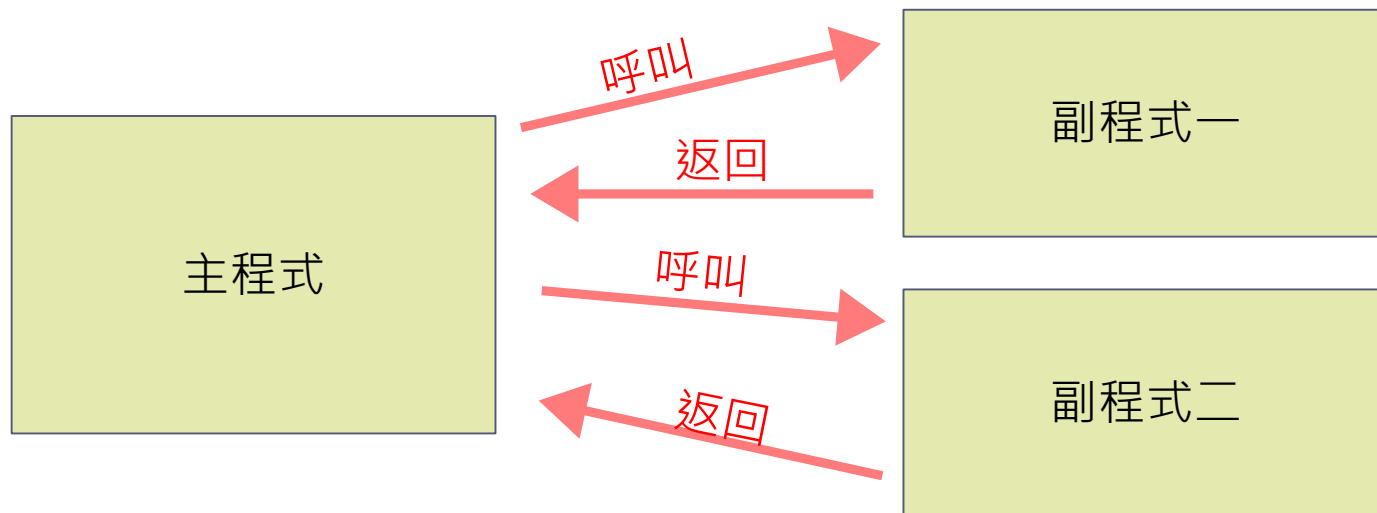
程序(副程式)與函數



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

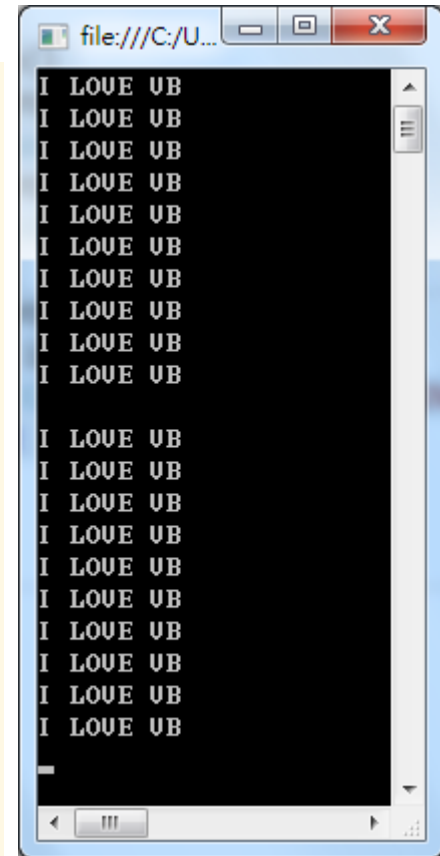
寫成副程式再呼叫

```
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
```

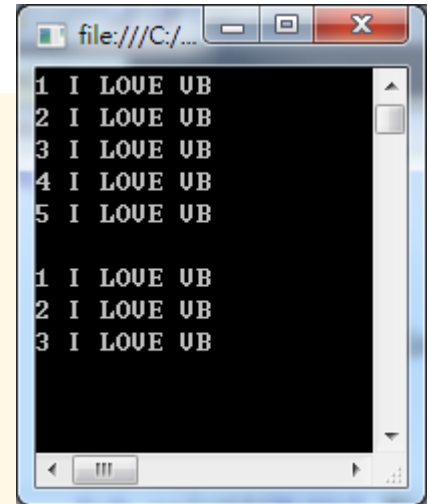


副程式(有傳遞參數)

- ▶ 將數值傳給副程式的X：

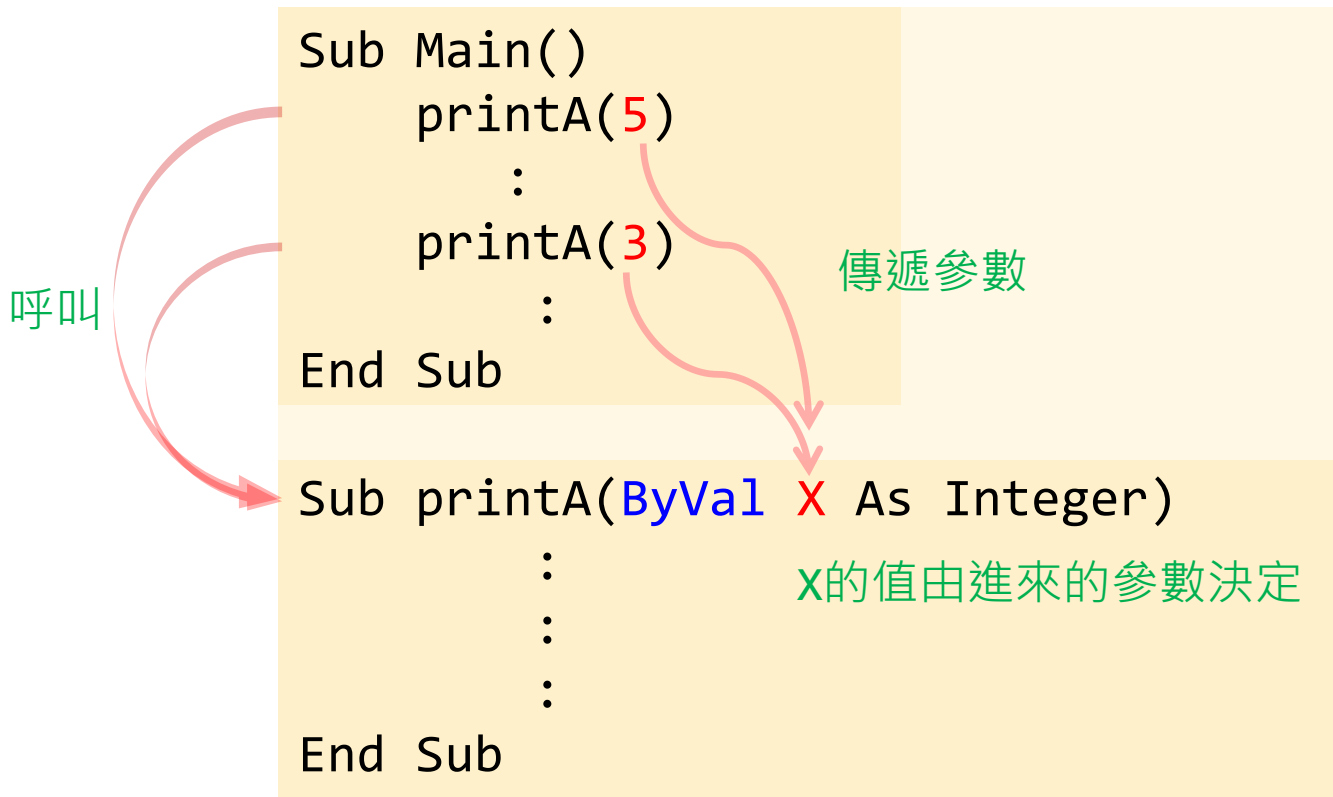
```
Sub Main()  
    printA(5)  
    Console.WriteLine()  
    printA(3)  
    Console.Read()  
End Sub
```

```
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



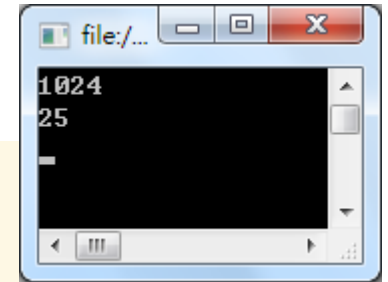
副程式(傳遞多個參數)

- ▶ 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
```



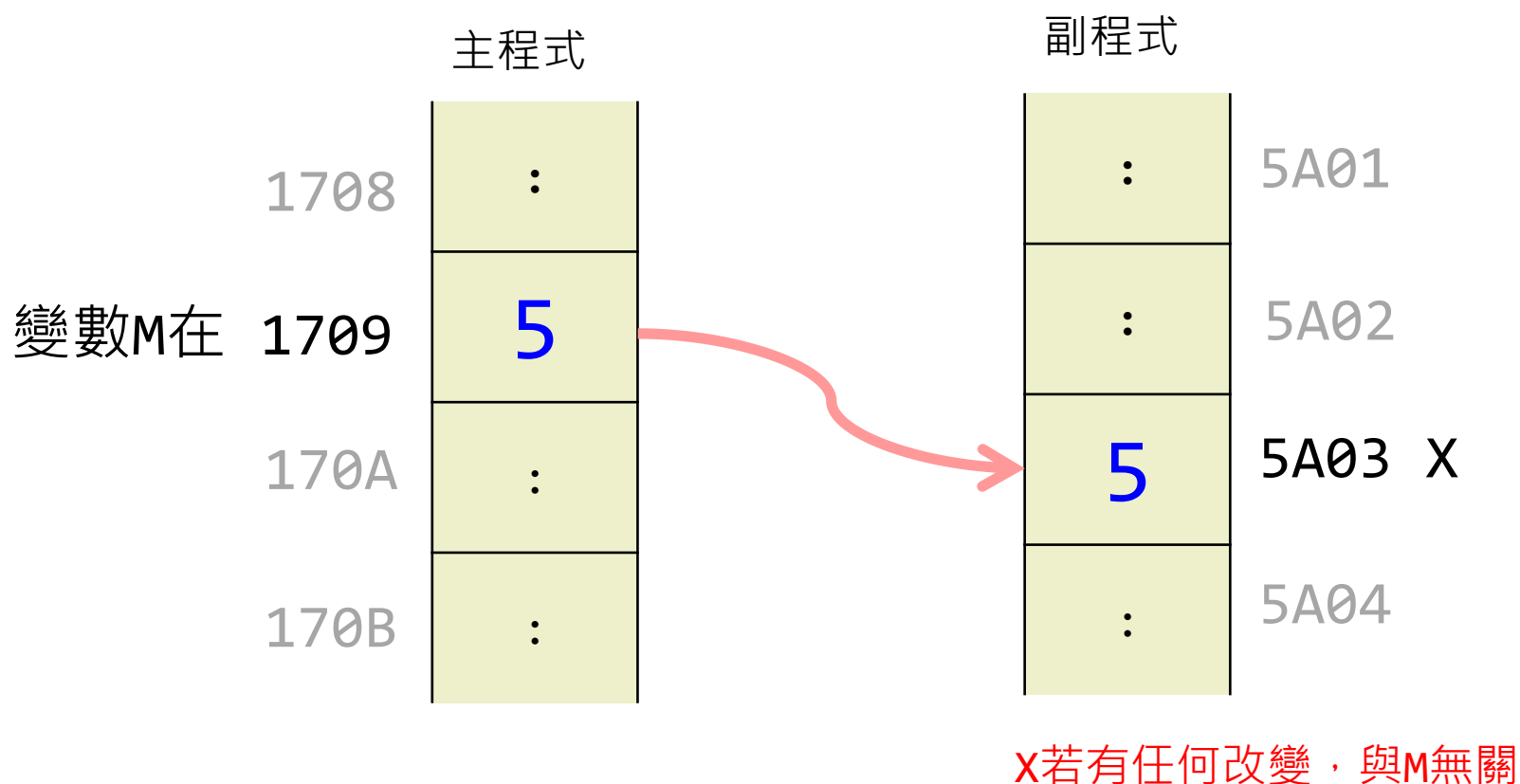
ByVal 與 ByRef

- ▶ 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

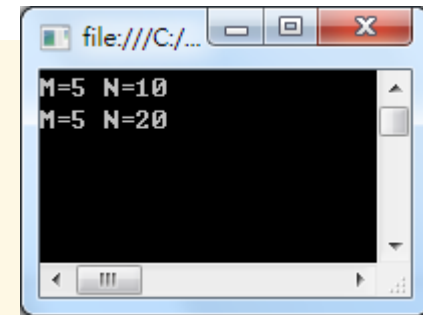


ByVal 與 ByRef

- ▶ M使用ByVal，N使用ByRef，可以看出N被副程式改變了

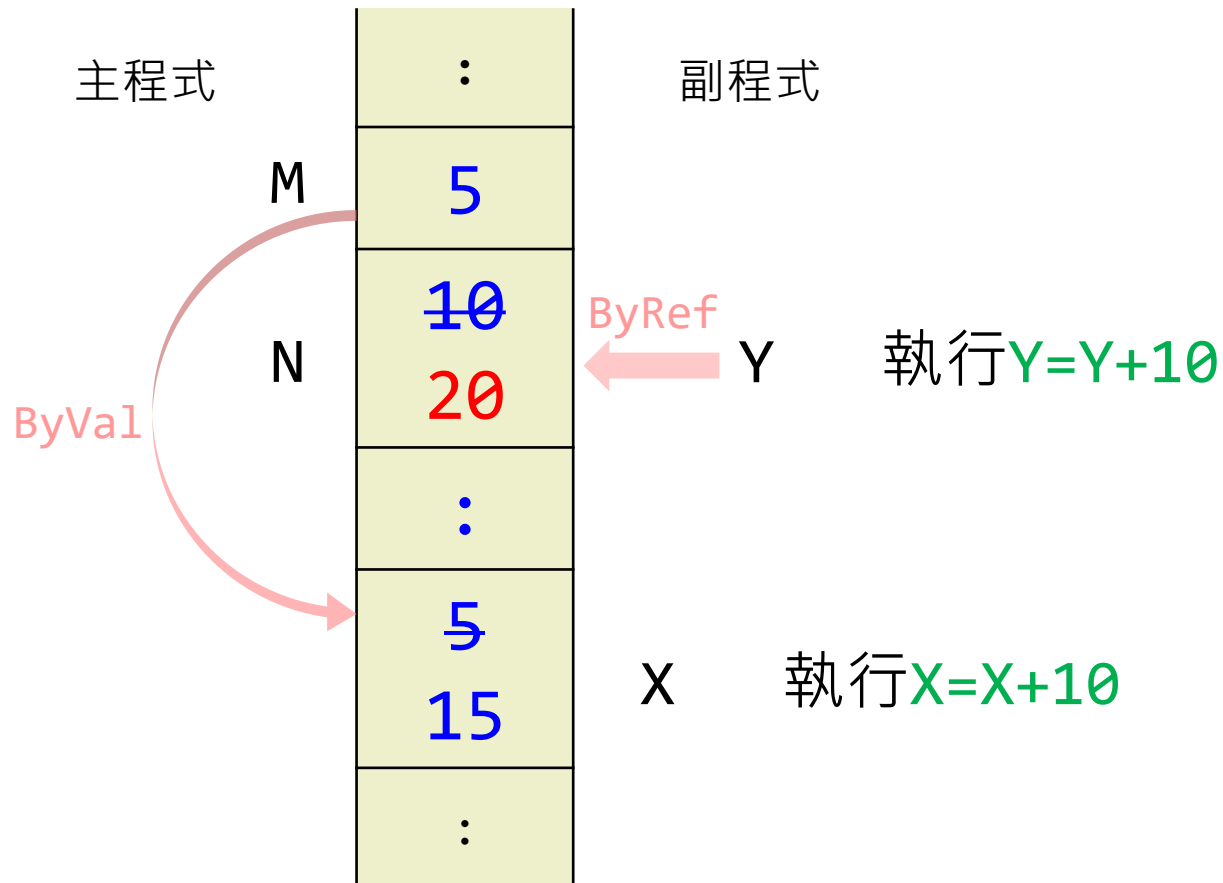
```
Sub Main()  
    Dim M, N As Integer  
    M = 5 : N = 10  
    Console.WriteLine("M=" & M & " N=" & N)  
    printA(M, N)  
    Console.WriteLine("M=" & M & " N=" & N)  
    Console.Read()  
End Sub
```

```
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
```

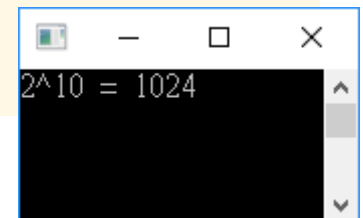
- 與副程式相同，參數可有可無
- 函數中執行到Return敘述則結束函數，並將指定的值傳回

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
```

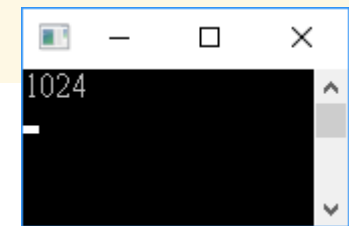


函數(Function)

▶ 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

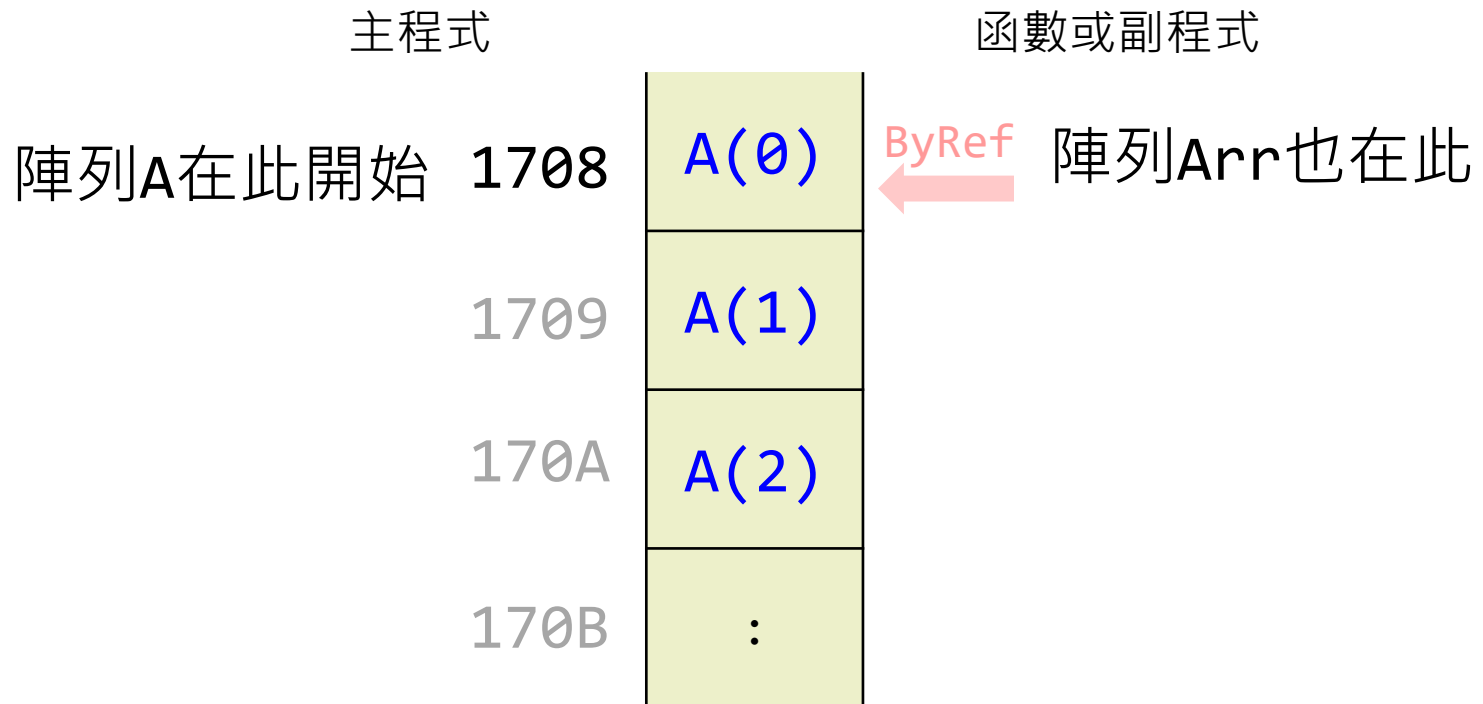
```
Function sqr(x, y)  
    sqr = x ^ y  
End Function
```

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

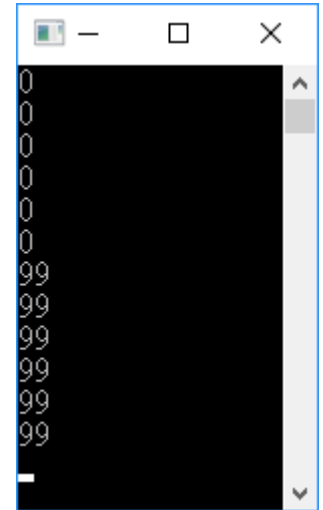
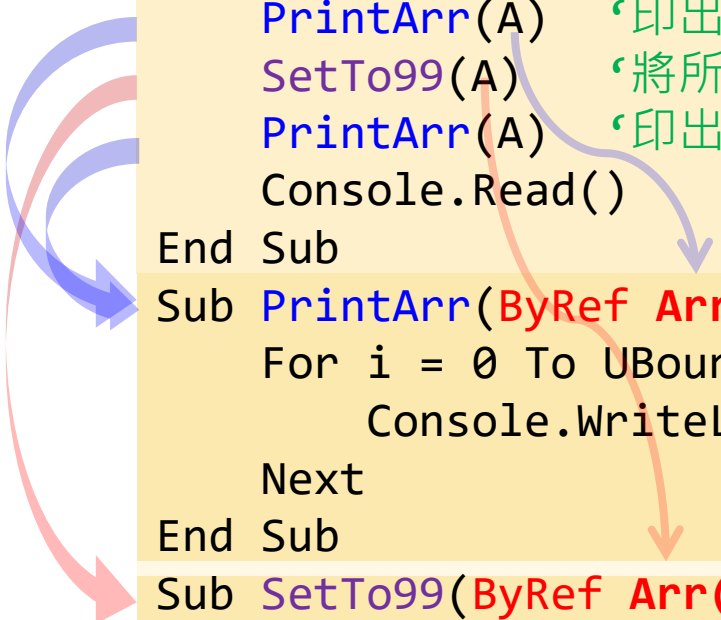
傳遞陣列

- ▶ 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

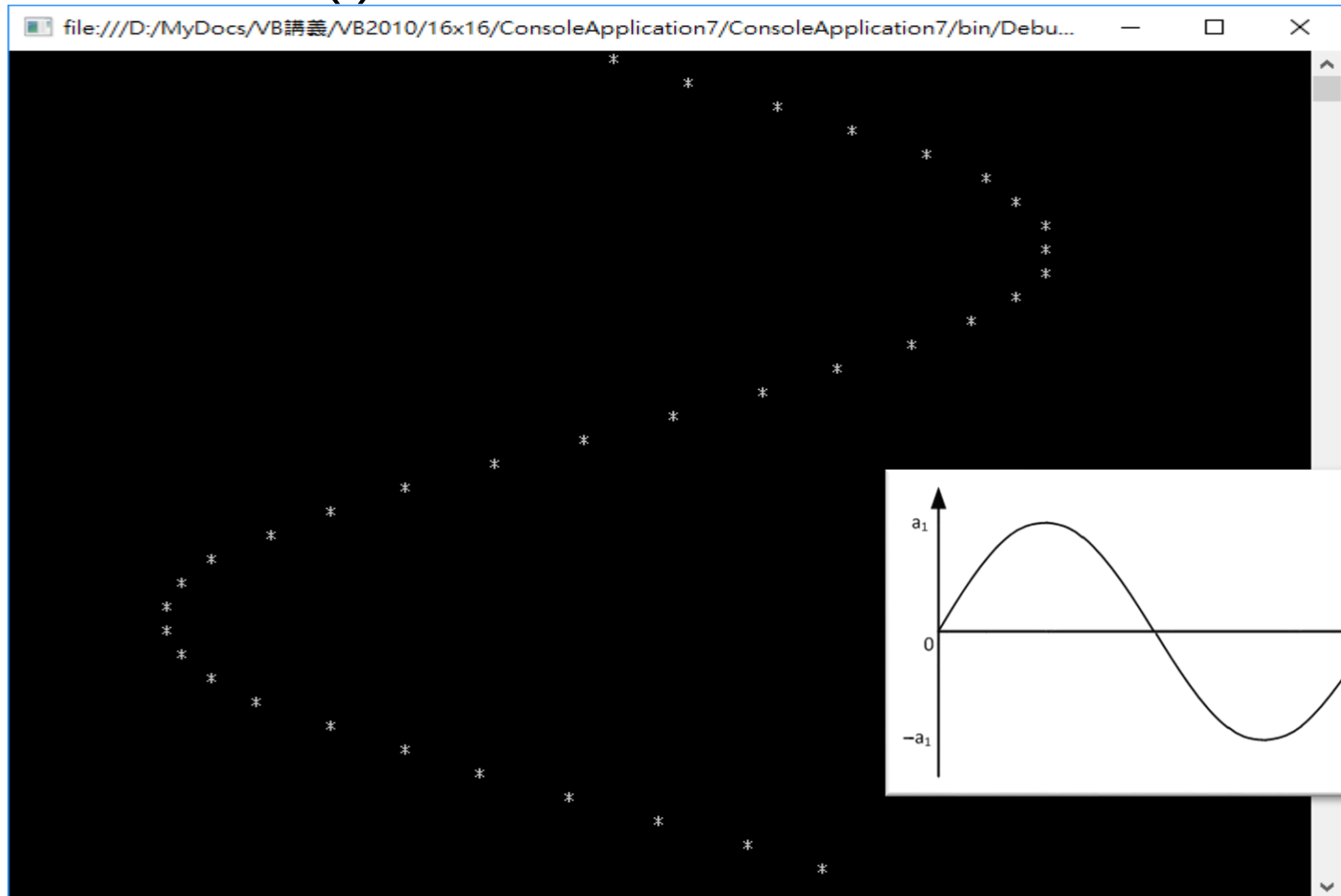
- ▶ 執行上述敘述後，控制權及交還主程式

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

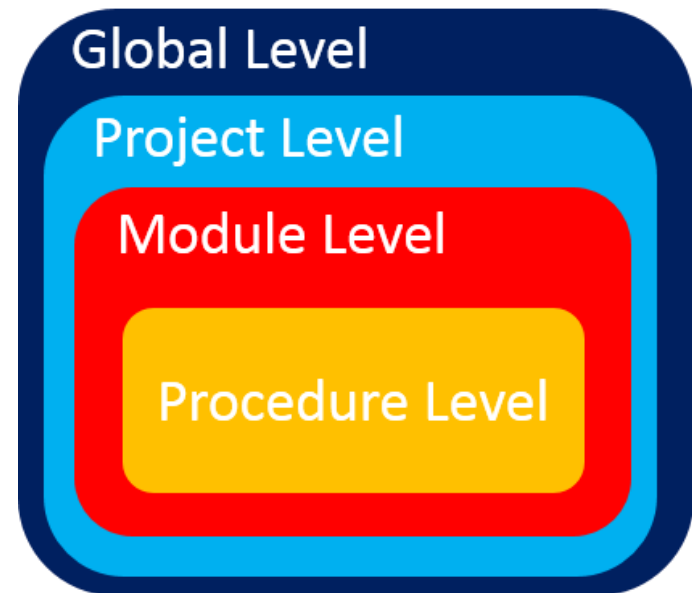
▶ 參考寫法：

```
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

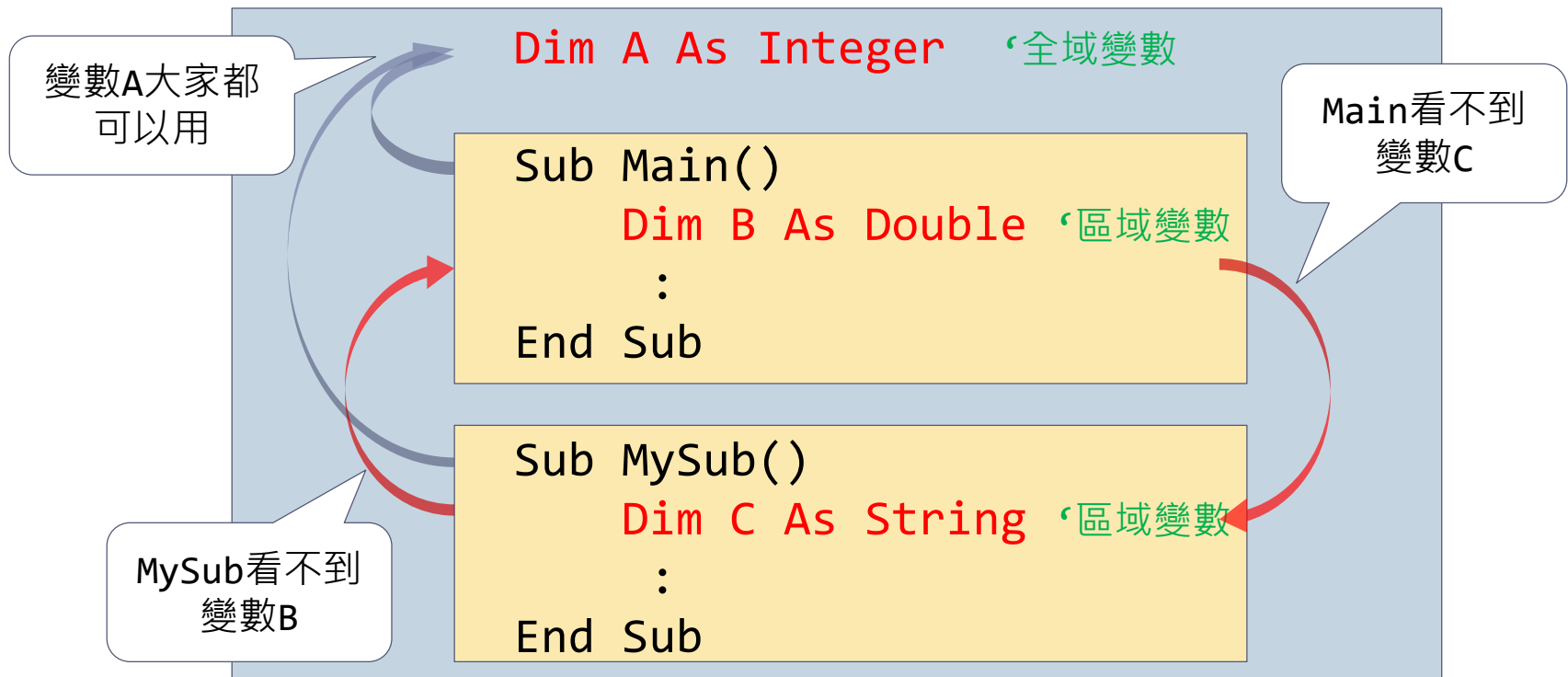
變數的可見範圍及生命週期

- ▶ 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



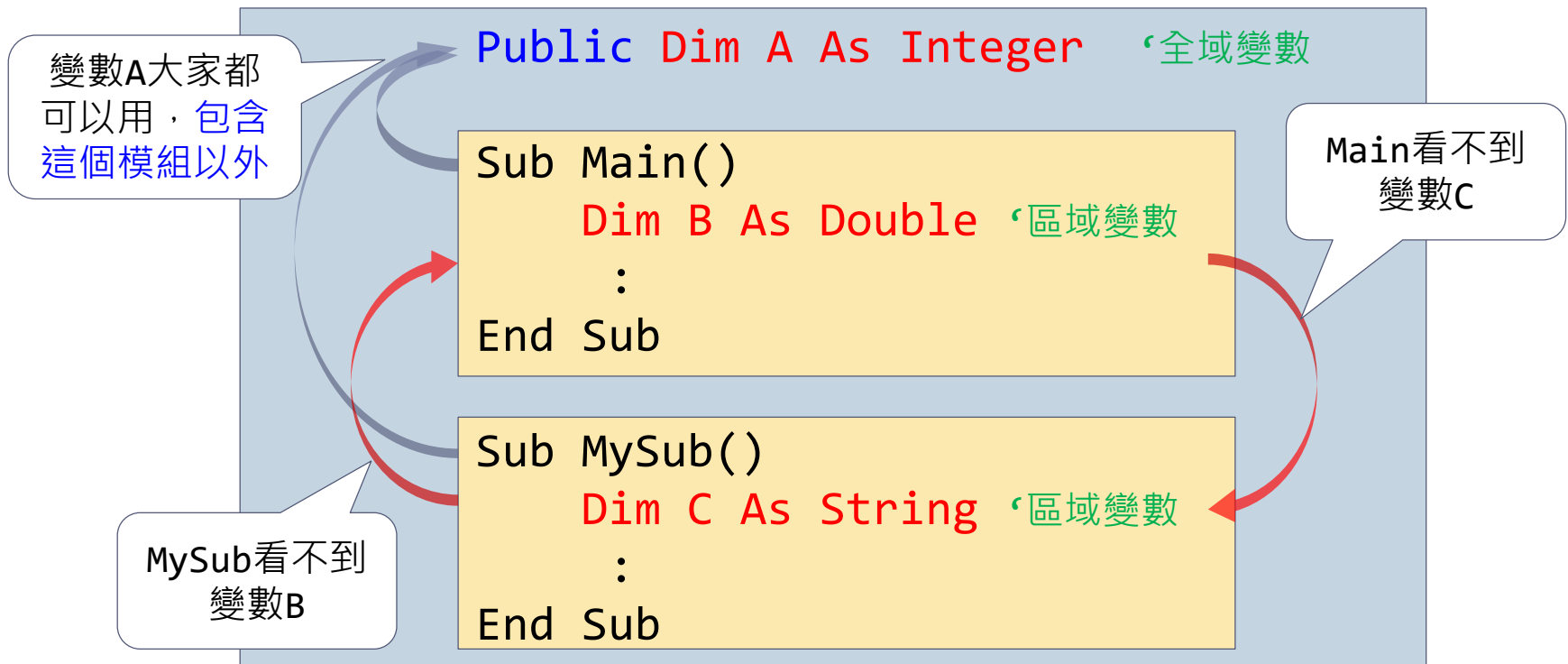
變數的可見範圍

- ▶ 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

Ch9.遞迴(recursive)

- ▶ 副程式可以被呼叫，當副程式裡又呼叫自己時，這種情況稱為遞迴。
- ▶ 有些問題用遞迴可以得到更簡潔的程式，但不一定會有高效率。
- ▶ 使用遞迴時要注意：
 - ▶ 1.傳入的參數：要能符合要解決的問題。
 - ▶ 2.終止條件：錯誤或沒有終止條件，將造成無盡迴圈甚至當機
- ▶ 經典題型：最大公因數 (GCD)、費波納西數列 (Fibonacci Sequence)、河內塔 (Hanoi Tower)、N 個字元的排列組合等。



Ch9.遞迴(recursive)

- ▶ 求 $1 + 2 + 3 + \dots + N$ 的和。

終止條件

結束呼叫，
回傳確定值

傳遞參數，呼叫自己

```
Function sum(N)
```

```
    Dim value As Integer
```

```
    If N = 1 Then
```

```
        value = 1
```

```
    Else
```

```
        value = N + sum(N - 1)
```

```
    End If
```

```
    Return value
```

```
End Function
```

```
Sub Main()
```

```
    Dim x, y As Integer
```

```
    Console.Write("計算1到N的總和，請輸入N:")
```

```
    x = Console.ReadLine
```

```
    y = sum(x)
```

```
    Console.WriteLine("1到" & x & "的總和為" & y)
```

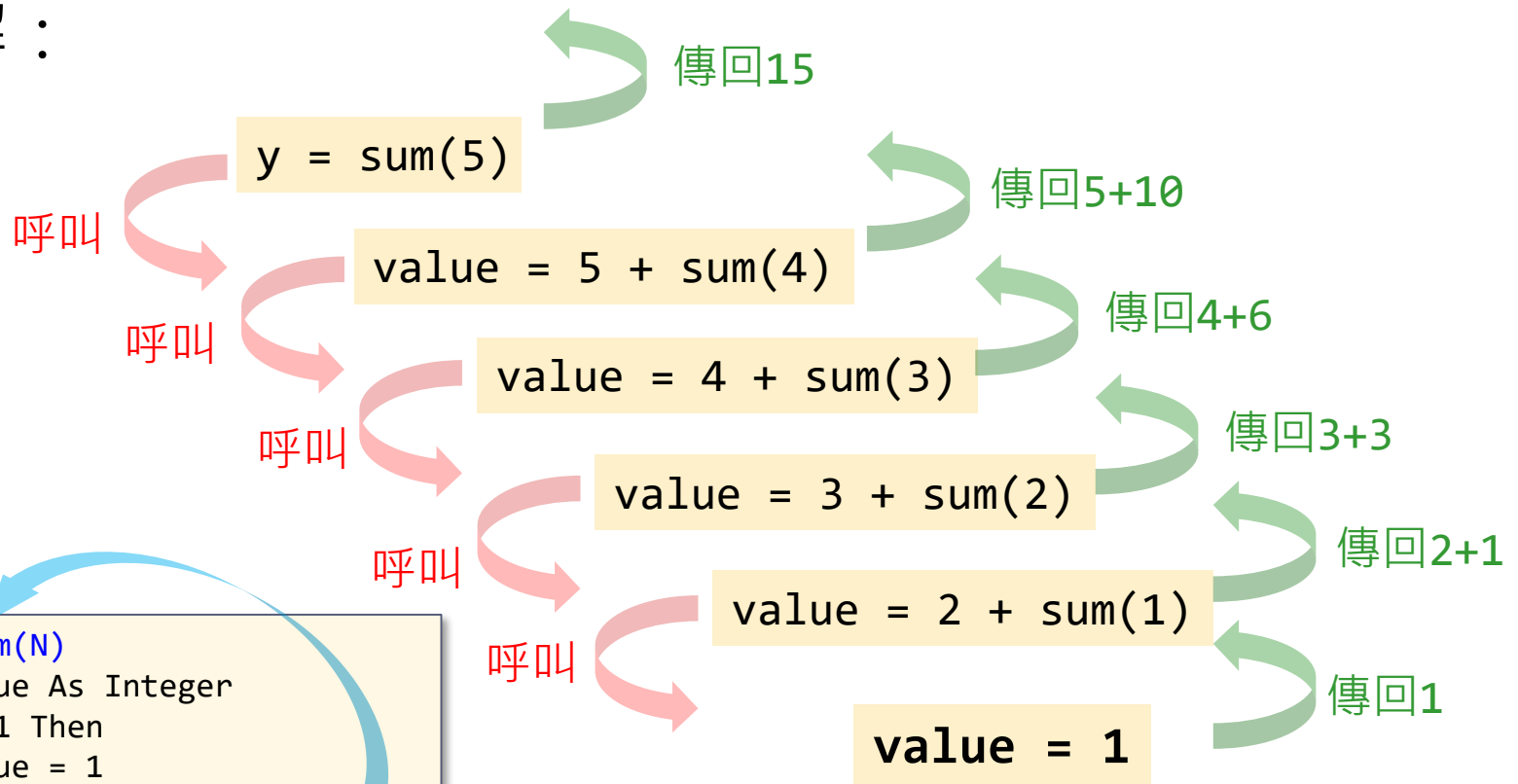
```
    Console.Read()
```

```
End Sub
```

- ▶ 假設輸入5

Ch9.遞迴(recursive)

► 圖解：



```
Function sum(N)
  Dim value As Integer
  If N = 1 Then
    value = 1
  Else
    value = N + sum(N - 1)
  End If
  Return value
End Function
```

Ch9.遞迴(recursive)

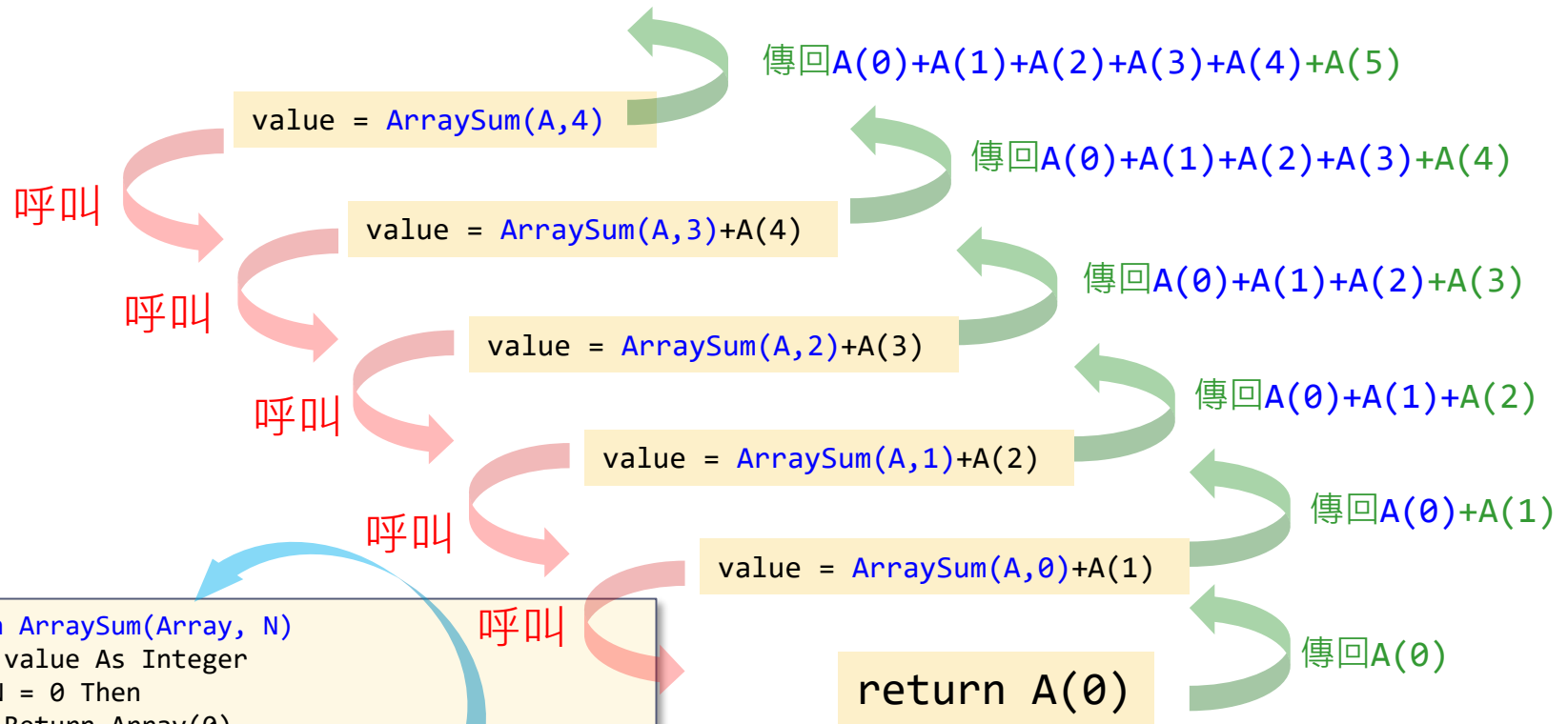
- ▶ 求陣列A中所有數值的和。

```
Function ArraySum(Array, N)
    Dim value As Integer
    If N = 0 Then
        Return Array(0)
    Else
        value = ArraySum(Array, N - 1) + Array(N)
    End If
    Return value
End Function

Sub Main()
    Dim A(), N As Integer
    Console.WriteLine("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1
        Console.WriteLine("請輸入A(" & i & ")的值:")
        A(i) = Console.ReadLine
    Next
    Console.WriteLine("Array內所有數值的總和為 " & ArraySum(A, N - 1))
    Console.Read()
End Sub
```

Ch9.遞迴(recursive)

► 圖解：(假設N=5)



```
Function ArraySum(Array, N)
  Dim value As Integer
  If N = 0 Then
    Return Array(0)
  Else
    value = ArraySum(Array, N - 1) + Array(N)
  End If
  Return value
End Function
```

Ch9.遞迴(recursive)

► 計算N!。

```
Function factorial(N)
    If N = 1 Then
        Return 1
    Else
        Return N * factorial(N - 1)
    End If
End Function

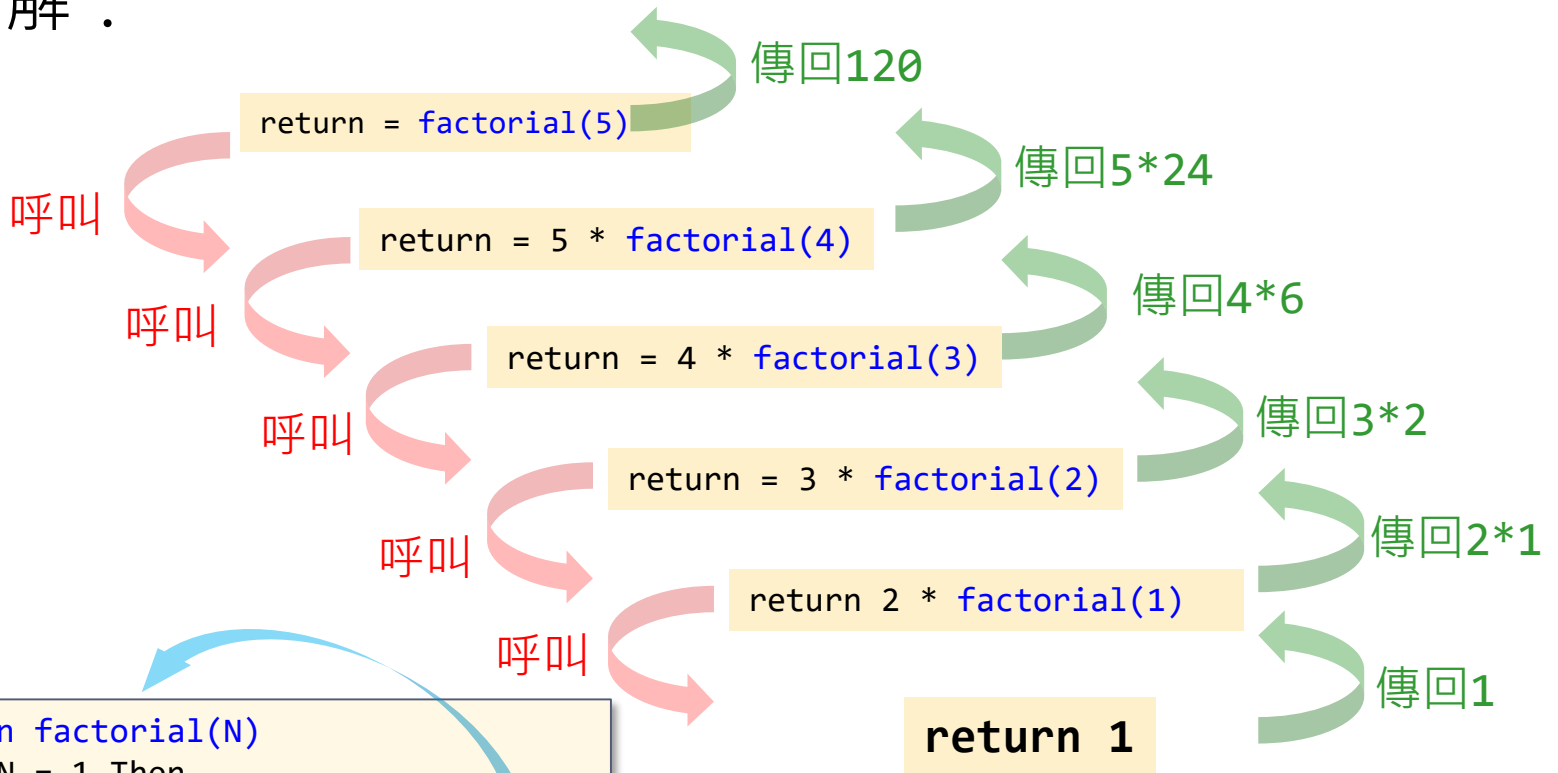
Sub Main()
    Dim x As Integer
    Console.Write("要計算幾階層?: ")
    x = Console.ReadLine
    Console.WriteLine(x & "! = " & factorial(x))
    Console.Read()
End Sub
```

► 假設輸入5



Ch9.遞迴(recursive)

► 圖解：



```
Function factorial(N)
  If N = 1 Then
    Return 1
  Else
    Return N * factorial(N - 1)
  End If
End Function
```

Ch9.遞迴(recursive)

► 求費氏數列F(n)。

```
Function F(N)
```

```
    Dim val As Integer
```

```
    If N = 1 Then
```

```
        Return 0
```

```
    ElseIf N = 2 Then
```

```
        Return 1
```

```
    Else
```

```
        val = F(N - 1) + F(N - 2)
```

```
    End If
```

```
    Return val
```

```
End Function
```

```
Sub Main()
```

```
    Dim x As Integer
```

```
    Console.Write("請輸入求F(N)的N為: ")
```

```
    x = Console.ReadLine
```

```
    Console.WriteLine("F(" & x & ") = " & F(x))
```

```
    Console.Read()
```

```
End Sub
```

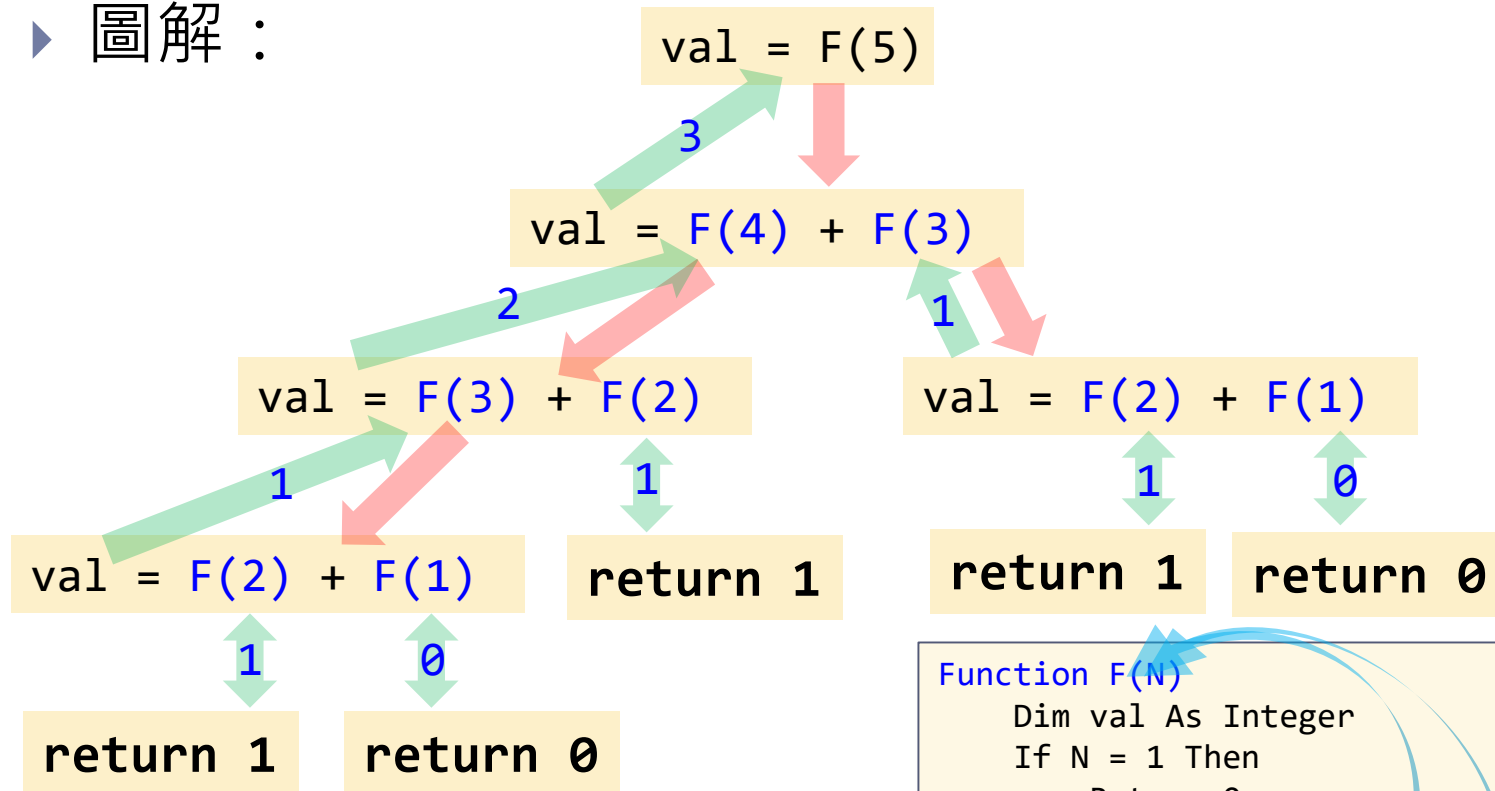
終止條件1

終止條件2

► 假設輸入5

Ch9.遞迴(recursive)

► 圖解：



► 最後傳回3

```
Function F(N)  
    Dim val As Integer  
    If N = 1 Then  
        Return 0  
    ElseIf N = 2 Then  
        Return 1  
    Else  
        val = F(N - 1) + F(N - 2)  
    End If  
    Return val  
End Function
```

Ch9. 習題

- ▶ 9-1. 以遞迴方式求 $1+b+b^2+\dots+b^n$ 。
- ▶ 9-2. 以遞迴方式求 $1\times 2+2\times 3+\dots+n(n+1)$ 。
- ▶ 9-3. 以遞迴方式求 $a+(a+b)+(a+2b)+\dots+(a+(n-1)b)$ 。
(注意 b 的係數在增加)



Ch9.遞迴(recursive)

- ▶ 9-1.以遞迴方式求 $1+b+b^2+\dots+b^n$ 。

```
Function nF(b, n)
    If n = 0 Then
        Return 1
    Else
        Return b ^ n + nF(b, n - 1)
    End If
End Function

Sub Main()
    Dim b, n As Integer
    Console.Write("請輸入基底b:")
    b = Console.ReadLine
    Console.Write("請輸入次方n:")
    n = Console.ReadLine
    Console.WriteLine("1+b+b^2+.....+b^n= " & nF(b, n))
    Console.Read()
End Sub
```

Ch9.遞迴(recursive)

- ▶ 9-2.以遞迴方式求 $1 \times 2 + 2 \times 3 + \dots + n(n+1)$ 。

```
Function f(n)
    If n = 2 Then
        Return 2
    Else
        Return n * (n - 1) + f(n - 1)
    End If
End Function

Sub Main()
    Dim n As Integer
    Console.Write("請輸入N:")
    n = Console.ReadLine
    Console.WriteLine("(1*2)+(2*3)+...+n*(n+1)= " & f(n + 1))
    Console.Read()
End Sub
```



Ch9.遞迴(recursive)

- ▶ 9-3.以遞迴方式求 $a + (a+b) + (a+2b) + \dots + (a+(n-1)b$ 。
(注意b的係數在增加)

```
Function f(a, b, n)
    If n = 0 Then
        Return a
    Else
        Return a + n * b + f(a, b, n - 1)
    End If
End Function

Sub Main()
    Dim a, b, n As Integer
    Console.Write("請輸入a:") : a = Console.ReadLine
    Console.Write("請輸入b:") : b = Console.ReadLine
    Console.Write("請輸入n:") : n = Console.ReadLine
    Console.WriteLine("結果為:" & f(a, b, n - 1))
    Console.Read()
End Sub
```

休息一下~

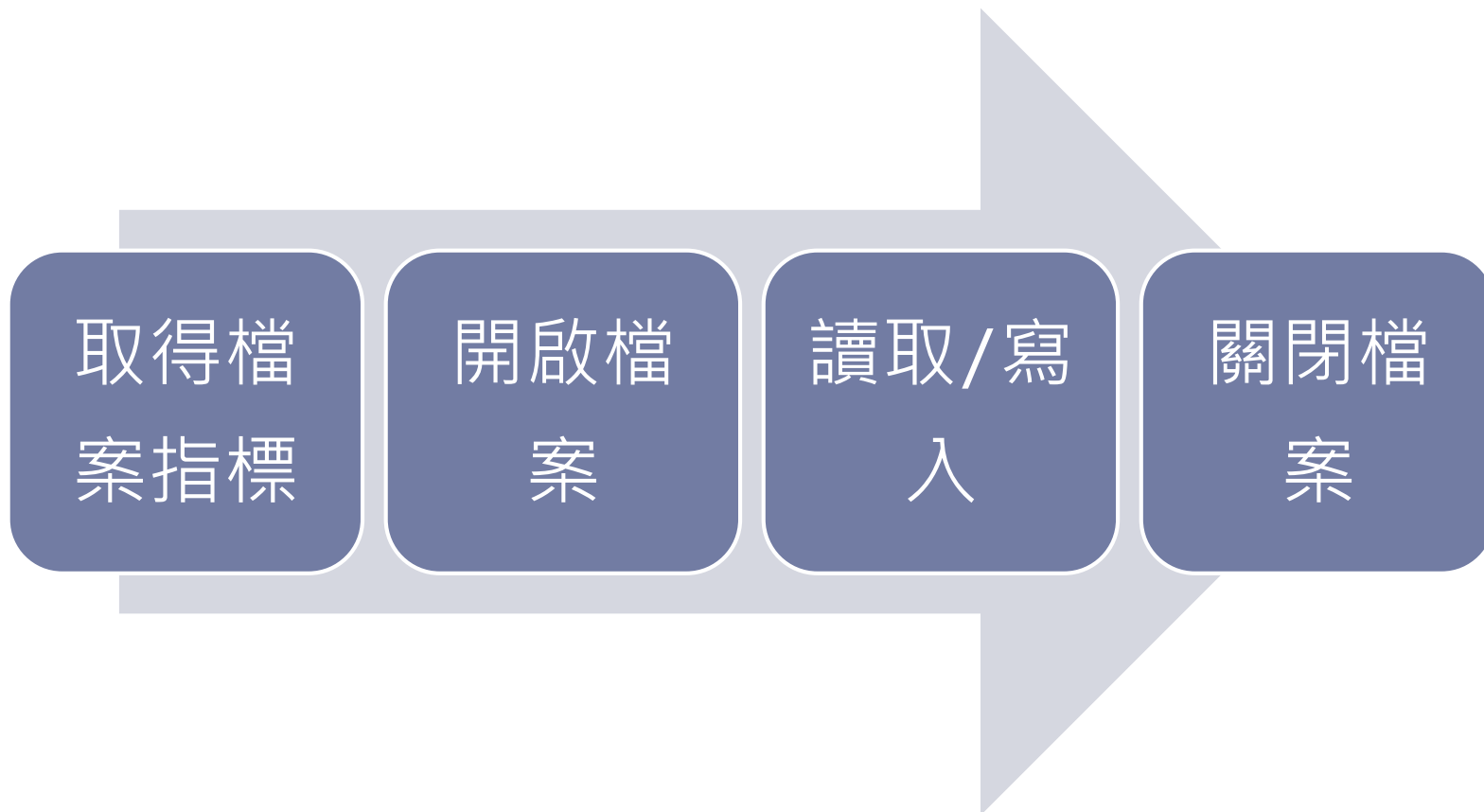


檔案讀寫



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”

Ch7.讀取檔案

- ▶ 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。
- ▶ 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

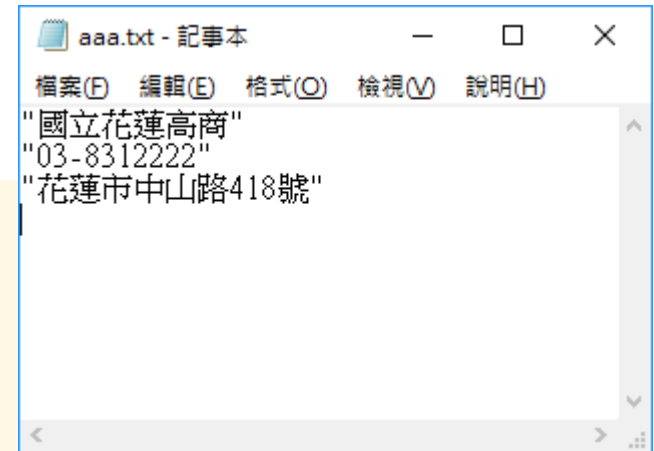
`EOF` (檔案指標)

- ▶ 當到達檔案結尾時傳回True，否則傳回False

檔案存取練習

► 寫入檔案

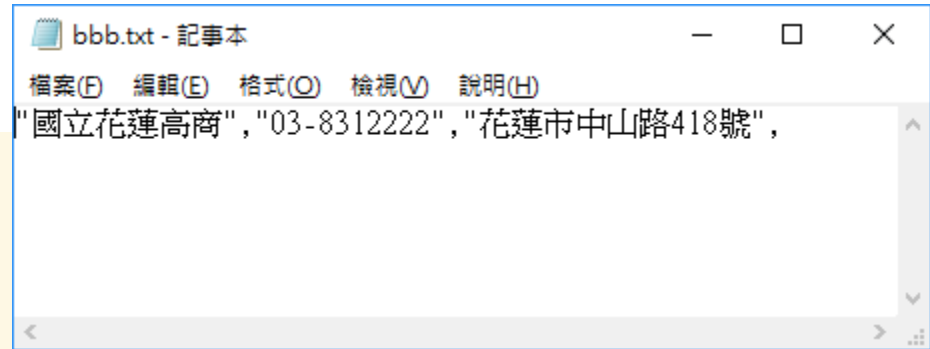
```
Dim fname As String
Dim fp As Integer
fname = "d:\aaa.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Output)
WriteLine(fp, "國立花蓮高商")
WriteLine(fp, "03-8312222")
WriteLine(fp, "花蓮市中山路418號")
FileClose(fp)
```



檔案存取練習

► 寫入檔案

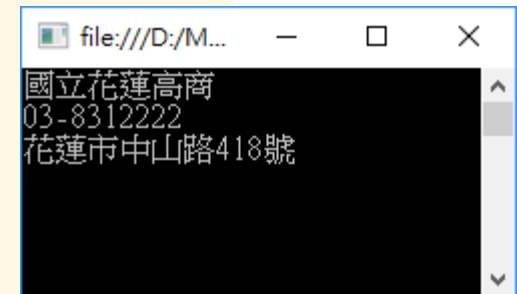
```
Dim fname As String
Dim fp As Integer
fname = "d:\bbb.txt"
fp = FreeFile()
FileOpen(fp, fname, OpenMode.Output)
Write(fp, "國立花蓮高商")
Write(fp, "03-8312222")
Write(fp, "花蓮市中山路418號")
FileClose(fp)
```



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

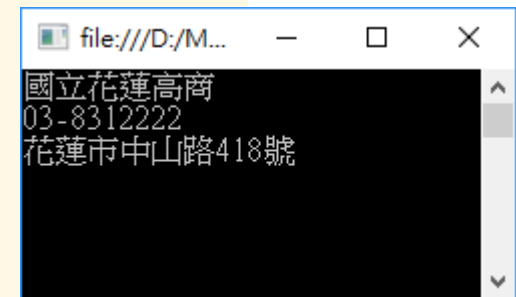
EOF (檔案指標)

- ▶ 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

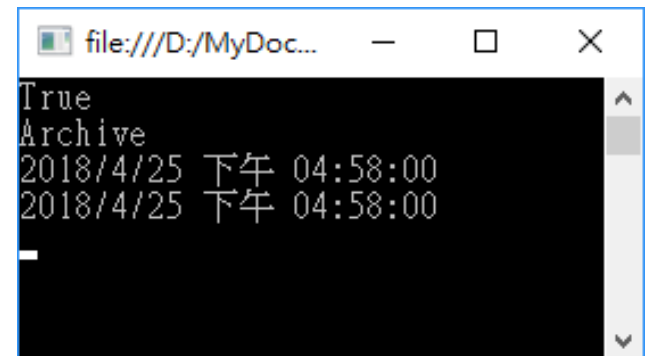
- ▶ 如果要讀取的檔案不存在怎麼辦？如果要寫入的檔案已經存在，不想覆蓋它？
- ▶ 使用File類別提供的一些方法來取得檔案資訊

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 console window. The title bar shows the file path "file:///D:/MyDoc...". The console output is as follows:

```
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
```

結構

- ▶ 又稱自訂型態(Structure)
- ▶ 當資料是由不同的資料型別組成時，可以使用結構將一個或多個資料型態包裝在一起，定義成一個新的形態。
- ▶ 李伯伯的書裡本章是介紹配置(configuration)，VB沒有這東西，但有點類似C或VB語言的結構(structure)。所以我們用結構來實作本章範例。



結構

- ▶ 自己定義新的複合資料當作一個新的形態

- ▶ 例如：

```
Structure student
    Dim name As String
    Dim age As Integer
    Dim BMI As Single
End Structure
```

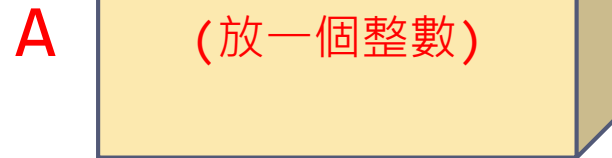
- ▶ 建立一個新的資料型態叫 student

```
Dim stu1 As student
John.name = "John Carter"
John.age = 20
John.BMI = 22.5
```

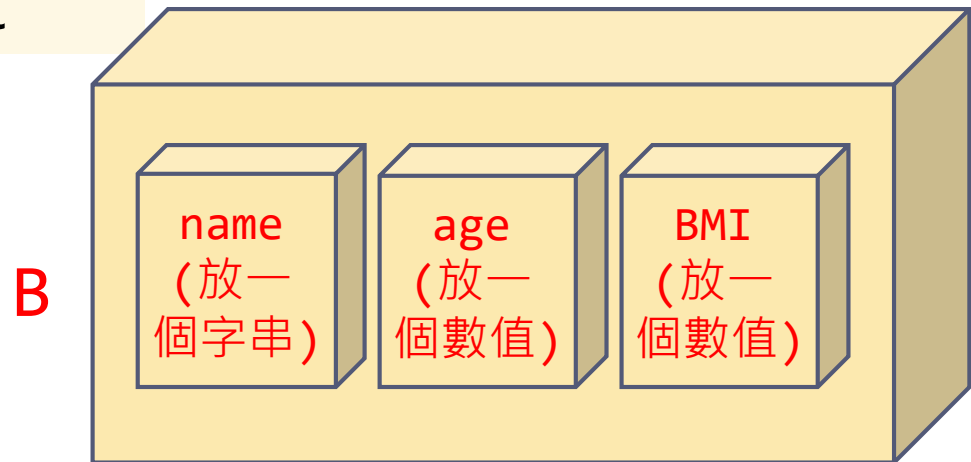
- ▶ 宣告stu1這個變數的型態是student，並用句點.來存取student內的成員

結構

- ▶ Dim A As Integer
- ▶ (單一變數)



- ▶ Dim B As student
- ▶ (結構變數)



結構

- ▶ 例如一個學生的課業資料如下：

```
name = 劉和師  
mathscore = 80  
chinesescore = 99  
engscore = 97
```

- ▶ 我們可以利用自行定義一個structure來結合這些資料，就叫studentdata好了，我們把它放在檔案內，檔名叫student_score.txt，我們可以很方便的來分析、搜尋資料。



結構

- ▶ 在磁碟內建立一個檔案，並寫出輸入並儲存班級學生各科成績等功能，內容如下。

學生姓名, 數學成績, 國文成績, 英文成績

劉和師,77,90,90
林志玲,67,76,57
劉德華,78,90,78

student_score.txt內容

結構

- ▶ 我們要做出以下各項功能：

```
----- 功能表 -----  
1. 新增一個學生的資料  
2. 查詢一個學生的資料  
3. 修改一個學生的資料  
4. 刪除一個學生的資料  
5. 印出所有學生的資料  
6. 求班級各科成績平均  
7. 結束程式  
-----  
請選擇：
```

- ▶ 為簡化程式，我們將每一個功能寫成一個副程式。
-

結構

- ▶ 先定義結構及全域變數：

```
' 定義資料格式
Structure studentdata
    Dim name As String           ' 學生姓名
    Dim mathscore As Single      ' 數學成績
    Dim chiscore As Single       ' 國文成績
    Dim engscore As Single       ' 英文成績
End Structure

Dim stu() As studentdata
```

- ▶ 要寫在所有的程式段之外(sub、Function、main等)，通常在程式一開始的地方。
-

結構

'讀入學生資料檔至結構陣列副程式

```
Sub read_student()  
    Dim f1, n As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\student_score.txt", OpenMode.Input)  
    n = 0  
    While Not EOF(f1)  
        n += 1  
        ReDim Preserve stu(n) '隨著讀入的資料筆數增加陣列  
        Input(f1, stu(n).name) '用 . 來存取結構內的成員  
        Input(f1, stu(n).mathscore)  
        Input(f1, stu(n).chiscore)  
        Input(f1, stu(n).engscore)  
    End While  
    FileClose(f1)  
End Sub
```



結構

' 將陣列資料寫入檔案

```
Sub write_to_file()  
    Dim f1 As Integer  
    f1 = FreeFile() 'Output模式會清除檔案裡原資料  
    FileOpen(f1, "d:\student_score.txt", OpenMode.Output)  
    For i = 1 To UBound(stu)  
        Write(f1, stu(i).name)  
        Write(f1, stu(i).mathscore)  
        Write(f1, stu(i).chiscore)  
        WriteLine(f1, stu(i).engscore) '最後一項資料加上換行  
    Next  
    FileClose(f1)  
End Sub
```



結構

► 主程式：

```
Sub Main()  
    Dim sel As Integer  
    read_student() '讀入檔案資料至結構陣列  
    While True  
        Console.WriteLine("----- 功能表 -----")  
        Console.WriteLine("1. 新增一個學生的資料")  
        Console.WriteLine("2. 查詢一個學生的資料")  
        Console.WriteLine("3. 修改一個學生的資料")  
        Console.WriteLine("4. 刪除一個學生的資料")  
        Console.WriteLine("5. 印出所有學生的資料")  
        Console.WriteLine("6. 求班級各科成績平均")  
        Console.WriteLine("7. 結束程式")  
        Console.WriteLine("-----")  
        Console.Write("請選擇:")  
        sel = Val(Console.ReadLine())  
        Select Case sel '每一項功能都去呼叫副程式  
            Case 1  
                input_studentdata()  
            Case 2  
                search_student()  
            Case 3  
                modify_student()  
            Case 4  
                remove_student()  
            Case 5  
                print_student()  
            Case 6  
                score_average()  
            Case 7  
                Exit While  
            Case Else  
                Console.WriteLine("錯誤，請輸入 1 ~ 7")  
        End Select  
    End While  
End Sub
```

結構

'新增一個學生的資料

```
Sub input_studentdata()  
    Dim name As String  
    Dim flag, i As Integer  
    Console.Write("請輸入學生姓名: ")  
    name = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(stu)  
        If stu(i).name = name Then  
            Console.WriteLine("該名學生已存在，無法新增")  
            flag = 1  
            Exit For  
        End If  
    Next  
    If flag = 0 Then  
        i = UBound(stu) + 1  
        ReDim Preserve stu(i)  
        stu(i).name = name  
        Console.Write("請輸入數學成績: ")  
        stu(i).mathscore = Console.ReadLine()  
        Console.Write("請輸入國文成績: ")  
        stu(i).chiscore = Console.ReadLine()  
        Console.Write("請輸入英文成績: ")  
        stu(i).engscore = Console.ReadLine()  
        write_to_file()  
        Console.WriteLine("新增學生" & name & "資料成功!")  
    End If  
End Sub
```

結構

·查詢一個學生的資料

```
Sub search_student()  
    Dim name As String  
    Dim flag As Integer  
    Console.Write("請輸入學生姓名: ")  
    name = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(stu)  
        If stu(i).name = name Then  
            Console.WriteLine("數學成績: " & stu(i).mathscore)  
            Console.WriteLine("國文成績: " & stu(i).chiscore)  
            Console.WriteLine("英文成績: " & stu(i).engscore)  
            flag = 1  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("無此學生")  
End Sub
```

結構

```
Sub modify_student() '修改一個學生的資料
    Dim name, temp As String
    Dim flag As Integer = 0
    Console.Write("請輸入學生姓名: ")
    name = Console.ReadLine()
    For i = 1 To UBound(stu)
        If stu(i).name = name Then
            Console.Write("原學生姓名: " & stu(i).name & ", 請輸入修改後姓名: ")
            temp = Console.ReadLine() '要考慮若不修改，直接按Enter的狀況
            If Len(temp) > 0 Then stu(i).name = temp
            Console.Write("原數學成績: " & stu(i).mathscore & ", 請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).mathscore = Val(temp)
            Console.Write("原國文成績: " & stu(i).chiscore & ", 請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).chiscore = Val(temp)
            Console.Write("原英文成績: " & stu(i).engscore & ", 請輸入修改後成績: ")
            temp = Console.ReadLine()
            If Len(temp) > 0 Then stu(i).engscore = Val(temp)
            write_to_file() '寫入檔案
            Console.WriteLine("修改學生" & name & "資料成功!")
            flag = 1 : Exit For
        End If
    Next
    If flag = 0 Then Console.WriteLine("無此學生")
End Sub
```

結構

```
Sub remove_student() '刪除一個學生的資料
    Dim name As String
    Dim flag As Integer = 0
    Console.Write("請輸入學生姓名: ")
    name = Console.ReadLine()
    For i = 1 To UBound(stu)
        If stu(i).name = name Then '找到資料後將後面的資料依序向前移動一個位置
            For j = i To UBound(stu) - 1
                stu(j).name = stu(j + 1).name '覆蓋掉要刪除的資料
                stu(j).mathscore = stu(j + 1).mathscore
                stu(j).chiscore = stu(j + 1).chiscore
                stu(j).engscore = stu(j + 1).engscore
            Next
            ReDim Preserve stu(UBound(stu) - 1)
            write_to_file() '寫入檔案
            Console.WriteLine("刪除學生" & name & "成功!")
            flag = 1
            Exit For
        End If
    Next
    If flag = 0 Then Console.WriteLine("無此學生")
End Sub
```

結構

· 印出所有學生的資料

```
Sub print_student()  
    Console.WriteLine("-----學生成績資料-----")  
    For i = 1 To UBound(stu)  
        Console.WriteLine("學生姓名: " & stu(i).name)  
        Console.WriteLine("數學成績: " & stu(i).mathscore)  
        Console.WriteLine("國文成績: " & stu(i).chiscore)  
        Console.WriteLine("英文成績: " & stu(i).engscore)  
        Console.WriteLine()  
    Next  
End Sub
```



結構

' 求班級各科成績平均

```
Sub score_average()  
    Dim math_sum, chi_sum, eng_sum As Single  
    Dim n As Integer  
    n = UBound(stu)  
    For i = 1 To n  
        math_sum += stu(i).mathscore  
        chi_sum += stu(i).chiscore  
        eng_sum += stu(i).engscore  
    Next  
    Console.WriteLine("班級數學平均: " & math_sum / n)  
    Console.WriteLine("班級國文平均: " & chi_sum / n)  
    Console.WriteLine("班級英文平均: " & eng_sum / n)  
End Sub
```



結構

- ▶ 設計一個簡易借還書程式。
 - ▶ 這個程式有以下功能：
 - ▶ 1.新增書籍
 - ▶ 2.借書
 - ▶ 3.還書
 - ▶ 4.結束程式
 - ▶ 結構的格式如下：
 - ▶ 1.書名
 - ▶ 2.出版社
 - ▶ 3.作者
 - ▶ 4.頁數
 - ▶ 5.借閱人(為none時表示尚無人借閱)
-



結構

- ▶ 程式執行後，檔案booklist.txt的內容參考。

資料依序為：書名,出版社,作者,頁數,借閱人

迷霧之子I,皇冠出版社,布蘭登山德勒,684,劉和師
哈利波特I,CROWN,JK羅琳,1532,none
西遊記,三民書局,吳承恩,1515,none

結構

- ▶ 先定義結構及全域變數：

```
'宣告全域變數
Dim books() As booklist

'定義資料格式
Structure booklist
    Dim title As String    '書名
    Dim com As String      '出版社
    Dim writer As String   '作者
    Dim page As Integer    '頁數
    Dim name As String     '借閱人
End Structure
```

結構

'讀入檔案資料

```
Sub readBook()
```

```
    Dim f1, n As Integer
```

```
    f1 = FreeFile()
```

```
    FileOpen(f1, "d:\booklist.txt", OpenMode.Input)
```

```
    n = 0
```

```
    While Not EOF(f1)
```

```
        n += 1
```

```
        ReDim Preserve books(n)
```

```
        Input(f1, books(n).title)
```

```
        Input(f1, books(n).com)
```

```
        Input(f1, books(n).writer)
```

```
        Input(f1, books(n).page)
```

```
        Input(f1, books(n).name)
```

```
    End While
```

```
    FileClose(f1)
```

```
End Sub
```

'隨著讀入的資料筆數增加陣列

'用 . 來存取結構內的成員



結構

```
' 寫入檔案資料
Sub writeBook()
    Dim f1 As Integer
    f1 = FreeFile()                'Output模式會清除檔案裡原資料
    FileOpen(f1, "d:\booklist.txt", OpenMode.Output)
    For i = 1 To UBound(books)
        Write(f1, books(i).title)
        Write(f1, books(i).com)
        Write(f1, books(i).writer)
        Write(f1, books(i).page)
        WriteLine(f1, books(i).name) '最後一項資料加上換行
    Next
    FileClose(f1)
End Sub
```



結構

```
Sub addBook() '新增書籍
    Dim title As String
    Dim flag, i As Integer
    Console.Write("請輸入欲新增書籍的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            Console.WriteLine("這本書已經存在了，無法新增")
            flag = 1 : Exit For
        End If
    Next
    If flag = 0 Then
        i = UBound(books) + 1
        ReDim Preserve books(i)
        books(i).title = title
        Console.Write("請輸入" & title & "的出版社: ") : books(i).com = Console.ReadLine()
        Console.Write("請輸入" & title & "的作者: ") : books(i).writer = Console.ReadLine()
        Console.Write("請輸入" & title & "的頁數: ") : books(i).page = Console.ReadLine()
        books(i).name = "none" '新增書本，尚無人借閱，故填入none
        writeBook()
        Console.WriteLine("新增書籍「" & title & "」資料成功!")
    End If
End Sub
```

結構

```
Sub borrowBook() '借書
    Dim title As String
    Dim flag As Integer
    Console.Write("請輸入欲借閱的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            flag = 1
            If books(i).name = "none" Then
                Console.Write("請輸入借閱人姓名:")
                books(i).name = Console.ReadLine()
                Console.WriteLine("借書成功")
                writeBook()
            Else
                Console.WriteLine("這本書已經被借走了")
            End If
        End If
    Exit For
    End If
Next
If flag = 0 Then Console.WriteLine("沒有這本書!")
End Sub
```

結構

```
Sub returnBook() '還書
    Dim title, name As String
    Dim flag As Integer
    Console.Write("請輸入歸還的書名: ")
    title = Console.ReadLine()
    flag = 0
    For i = 1 To UBound(books)
        If books(i).title = title Then
            flag = 1
            Console.Write("請輸入借閱人姓名:")
            name = Console.ReadLine()
            If books(i).name = name Then
                books(i).name = "none"
                Console.WriteLine("還書成功")
                writeBook()
            Else
                Console.WriteLine(title & "不是被" & name & "所借走的!")
            End If
        End If
    Exit For
    End If
Next
If flag = 0 Then Console.WriteLine("沒有這本書!")
End Sub
```

結構

' 印出所有書籍資料

```
Sub listBook()  
    Console.WriteLine("-----所有書籍資料-----")  
    For i = 1 To UBound(books)  
        Console.WriteLine("書名: " & books(i).title)  
        Console.WriteLine("出版社: " & books(i).com)  
        Console.WriteLine("作者: " & books(i).writer)  
        Console.WriteLine("頁數: " & books(i).page)  
        Console.WriteLine("借閱人: " & books(i).name)  
        Console.WriteLine()  
    Next  
End Sub
```



結構

```
Sub Main() '主程式
    Dim sel As Integer
    readBook() '讀入檔案資料至結構陣列
    While True
        Console.WriteLine("----- 功能表 -----")
        Console.WriteLine("1. 新增書籍")
        Console.WriteLine("2. 借書")
        Console.WriteLine("3. 還書")
        Console.WriteLine("4. 印出所有書籍資料")
        Console.WriteLine("5. 結束程式")
        Console.WriteLine("-----")
        Console.Write("請選擇:")
        sel = Val(Console.ReadLine())
        Select Case sel '每一項功能都去呼叫副程式
            Case 1
                addBook()
            Case 2
                borrowBook()
            Case 3
                returnBook()
            Case 4
                listBook()
            Case 5
                Exit While
            Case Else
                Console.WriteLine("錯誤，請輸入 1 ~ 5")
        End Select
    End While
End Sub
```

結構練習

- ▶ 我們要替一家醫院處理一些病患資料如下：
 - ▶ 1.健保號碼
 - ▶ 2.姓名
 - ▶ 3.性別
 - ▶ 4.電話
 - ▶ 5.醫療科別
 - ▶ 寫一程式，以結構的方式，輸入並儲存病人資料，要有以下功能：
 - ▶ 1. 新增一位病人資料
 - ▶ 2. 修改一位病人資料
 - ▶ 3. 查詢一位病人資料
 - ▶ 4. 刪除一位病人資料
 - ▶ 5. 列印所有女性病人姓名
 - ▶ 6. 結束程式
-



結構練習

- ▶ 程式執行後，檔案patientlist.txt的內容參考。

資料依序為：健保號碼,姓名,性別,電話,醫療科別

1001,張三,男,03-8312225,內科
1002,李四,男,03-8312227,外科
1003,隋棠,女,02-32569874,婦科
1004,林志玲,女,02-98745632,家醫科

結構練習-參考程式

'宣告全域變數

```
Dim patients() As patientlist
```

'定義資料格式

```
Structure patientlist
```

```
    Dim PID As String          '健保號碼
```

```
    Dim name As String        '姓名
```

```
    Dim sex As String          '性別
```

```
    Dim tel As String          '電話
```

```
    Dim division As String     '醫療科別
```

```
End Structure
```



結構練習-參考程式

'讀入檔案資料

```
Sub readPatient()  
    Dim f1, n As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\patientlist.txt", OpenMode.Input)  
    n = 0  
    While Not EOF(f1)  
        n += 1  
        ReDim Preserve patients(n) '隨著讀入的資料筆數增加陣列  
        Input(f1, patients(n).PID) '用 . 來存取結構內的成員  
        Input(f1, patients(n).name)  
        Input(f1, patients(n).sex)  
        Input(f1, patients(n).tel)  
        Input(f1, patients(n).division)  
    End While  
    FileClose(f1)  
End Sub
```

結構練習-參考程式

'寫入檔案資料

```
Sub writePatient()  
    Dim f1 As Integer  
    f1 = FreeFile()  
    FileOpen(f1, "d:\patientlist.txt", OpenMode.Output)  
    For i = 1 To UBound(patients)  
        Write(f1, patients(i).PID)  
        Write(f1, patients(i).name)  
        Write(f1, patients(i).sex)  
        Write(f1, patients(i).tel)  
        WriteLine(f1, patients(i).division)  
    Next  
    FileClose(f1)  
End Sub
```



結構練習-參考

'新增一位病人資料

```
Sub addPatient()  
    Dim PID As String  
    Dim flag, i As Integer  
    Console.WriteLine("請輸入新病患的健保號碼: ")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            Console.WriteLine("本病患資料已存在了")  
            flag = 1  
            Exit For  
        End If  
    Next  
    If flag = 0 Then  
        i = UBound(patients) + 1  
        ReDim Preserve patients(i)  
        patients(i).PID = PID  
        Console.WriteLine("請輸入" & PID & "的姓名: ")  
        patients(i).name = Console.ReadLine()  
        Console.WriteLine("請輸入" & PID & "的性別: ")  
        patients(i).sex = Console.ReadLine()  
        Console.WriteLine("請輸入" & PID & "的電話: ")  
        patients(i).tel = Console.ReadLine()  
        Console.WriteLine("請輸入" & PID & "的醫療科別: ")  
        patients(i).division = Console.ReadLine()  
        writePatient()  
        Console.WriteLine("新增成功!")  
    End If  
End Sub
```

結構練習-參考程式

'修改一位病人資料

```
Sub editPatient()  
    Dim PID, name, sex, tel, division As String  
    Dim flag As Integer  
    Console.Write("請輸入欲修改的病患的健保號碼: ")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            flag = 1  
            Console.WriteLine("請輸入新資料，空白則不修改")  
            Console.Write("原姓名:" & patients(i).name & ",新姓名:")  
            name = Console.ReadLine()  
            If name <> "" Then patients(i).name = name  
            Console.Write("原性別:" & patients(i).sex & ",新性別:")  
            sex = Console.ReadLine()  
            If sex <> "" Then patients(i).sex = sex  
            Console.Write("原電話:" & patients(i).tel & ",新電話:")  
            tel = Console.ReadLine()  
            If tel <> "" Then patients(i).tel = tel  
            Console.Write("原醫療科別:" & patients(i).division & ",新醫療科別:")  
            division = Console.ReadLine()  
            If division <> "" Then patients(i).division = division  
            writePatient()  
            Console.WriteLine("資料修改完畢!")  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("查無本病患資料")  
End Sub
```

結構練習-參考程式

'查詢一位病人資料

```
Sub searchPatient()  
    Dim PID As String  
    Dim flag As Integer  
    Console.WriteLine("請輸入欲查詢病患的健保號碼：")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            flag = 1  
            Console.WriteLine("-----病歷號碼" & PID & "的資料-----")  
            Console.WriteLine("姓名：" & patients(i).name)  
            Console.WriteLine("性別：" & patients(i).sex)  
            Console.WriteLine("電話：" & patients(i).tel)  
            Console.WriteLine("醫療科別：" & patients(i).division)  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("查無本病患資料!")  
End Sub
```

結構練習-參考程式

'刪除一位病人資料

```
Sub deletePatient()  
    Dim PID As String  
    Dim flag As Integer  
    Console.Write("請輸入欲刪除病患的健保號碼: ")  
    PID = Console.ReadLine()  
    flag = 0  
    For i = 1 To UBound(patients)  
        If patients(i).PID = PID Then  
            For j = i To UBound(patients) - 1  
                patients(j).PID = patients(j + 1).PID '覆蓋掉要刪除的資料  
                patients(j).name = patients(j + 1).name  
                patients(j).sex = patients(j + 1).sex  
                patients(j).tel = patients(j + 1).tel  
                patients(j).division = patients(j + 1).division  
            Next  
            ReDim Preserve patients(UBound(patients) - 1)  
            writePatient()  
            Console.WriteLine("本筆資料已刪除!")  
            Exit For  
        End If  
    Next  
    If flag = 0 Then Console.WriteLine("查無本病患資料!")  
End Sub
```

結構練習-參考程式

'列印所有女性病人姓名

```
Sub printFemalePatient()  
    Console.WriteLine("-----所有女性病人姓名-----")  
    For i = 1 To UBound(patients)  
        If patients(i).sex = "女" Then  
            Console.WriteLine("健保號碼: " & patients(i).PID &  
                               ", 姓名: " & patients(i).name)  
        End If  
    Next  
End Sub
```



結構練習

參考程式：

```
Sub Main() '主程式
    Dim sel As Integer
    readPatient() '讀入檔案資料至結構陣列
    While True
        Console.WriteLine("----- 功能表 -----")
        Console.WriteLine("1. 新增一位病人資料")
        Console.WriteLine("2. 修改一位病人資料")
        Console.WriteLine("3. 查詢一位病人資料")
        Console.WriteLine("4. 刪除一位病人資料")
        Console.WriteLine("5. 列印所有女性病人姓名")
        Console.WriteLine("6. 結束程式")
        Console.WriteLine("-----")
        Console.Write("請選擇:")
        sel = Val(Console.ReadLine())
        Select Case sel '每一項功能都去呼叫副程式
            Case 1
                addPatient()
            Case 2
                editPatient()
            Case 3
                searchPatient()
            Case 4
                deletePatient()
            Case 5
                printFemalePatient()
            Case 6
                Exit While
            Case Else
                Console.WriteLine("錯誤，請輸入 1 ~ 6")
        End Select
    End While
End Sub
```

休息一下~

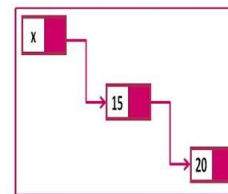


資料結構

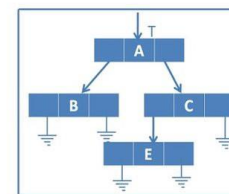
- ▶ 資料結構(Data Structure)是指電腦中儲存、組織資料的方式。
- ▶ 解決電腦問題時，使用適當的資料結構可以簡化程式，讓事情變簡單也更有效率。



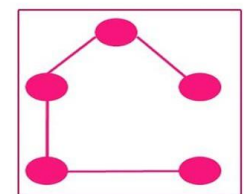
Sorting



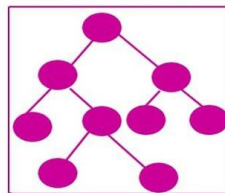
Link list



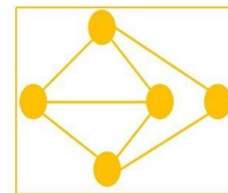
list



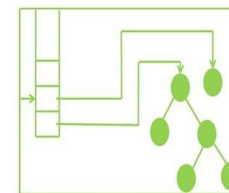
spanning tree



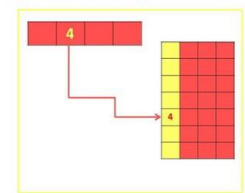
Tree



Graph



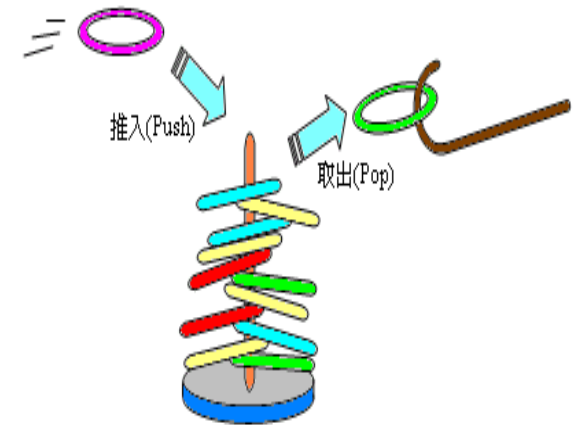
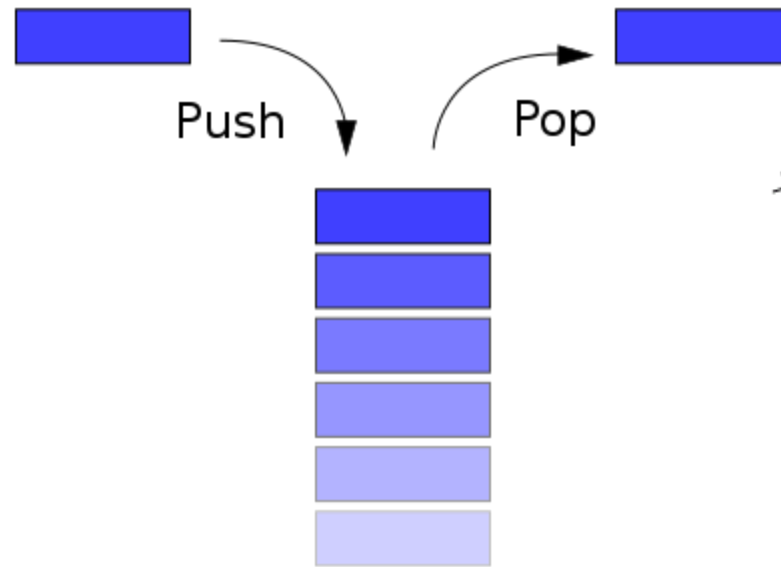
Stack



Hashing

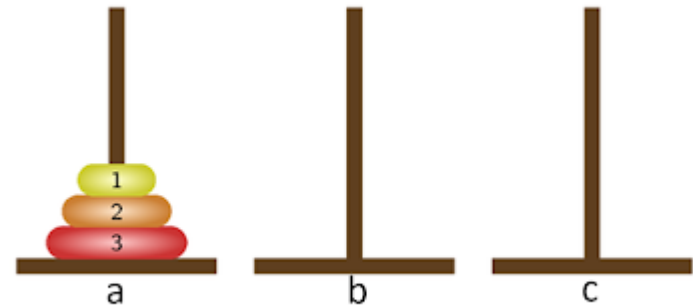
堆疊(Stack)

- ▶ 在日常生活中，把物品(如餐盤、書本)由桌面一個一個向上疊放，取用時由最上面一個個向下拿去，這種觀念稱為堆疊(stack)
- ▶ 存入稱**PUSH**，取出稱**POP**



堆疊(Stack)

- ▶ 堆疊觀念：先進後出(First In Last Out, **FILO**)或稱為後進先出(Last In First Out, **LIFO**)
- ▶ 堆疊觀念在計算機中使用很廣，例如主副程式間訊息傳送，CPU中斷處理、遞迴程式的呼叫及返還
- ▶ 最著名的問題就是：河內塔



堆疊(Stack)

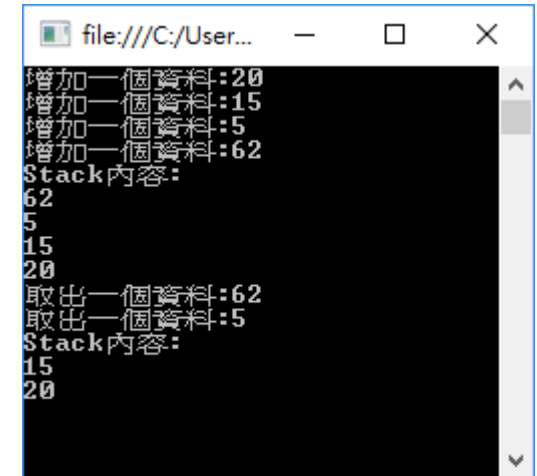
- ▶ 堆疊可以用陣列實現，也可以用內建的堆疊物件
- ▶ 堆疊物件常用的方法如下：

名稱	說明
Count	取得堆疊中目前的項目數量
Pop	從堆疊最頂端取出一個項目，同時將其移除
Push	從堆疊最頂端加入一個項目
Peek	從堆疊最頂端取出一個項目，但不移除

堆疊(Stack)

► 使用陣列：

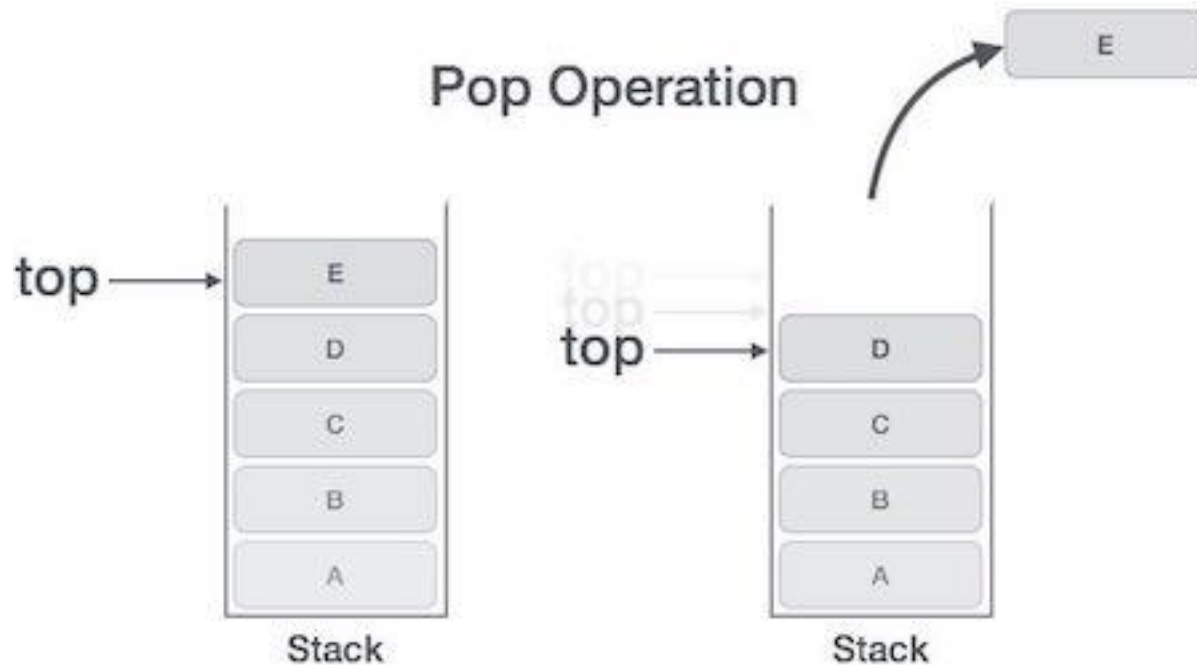
```
Dim S(0) As Integer
Sub Main()
    PUSH(20) : PUSH(15) : PUSH(5) : PUSH(62)
    PrintS()
    POP() : POP()
    PrintS()
    Console.Read()
End Sub
Sub PUSH(x) '存入資料，陣列大小動態增加
    ReDim Preserve S(UBound(S) + 1)
    S(UBound(S)) = x
    Console.WriteLine("增加一個資料:" & x)
End Sub
Sub POP() '取出資料，要注意是否已空
    If UBound(S) >= 1 Then
        Console.WriteLine("取出一個資料:" & S(UBound(S)))
        ReDim Preserve S(UBound(S) - 1)
    Else
        Console.WriteLine("Stack是空的")
    End If
End Sub
Sub PrintS() '印出堆疊內容
    Console.WriteLine("Stack內容:")
    For i = UBound(S) To 1 Step -1
        Console.WriteLine(S(i))
    Next
End Sub
```



```
file:///C:/User...
增加一個資料:20
增加一個資料:15
增加一個資料:5
增加一個資料:62
Stack內容:
62
5
15
20
取出一個資料:62
取出一個資料:5
Stack內容:
15
20
```

堆疊(Stack)

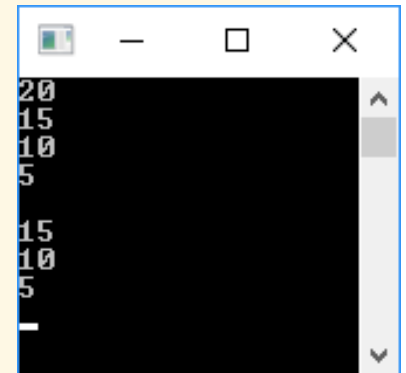
- 堆疊使用動態陣列的概念比較簡單，在陣列最後增加或減少一個元素即可



堆疊(Stack)

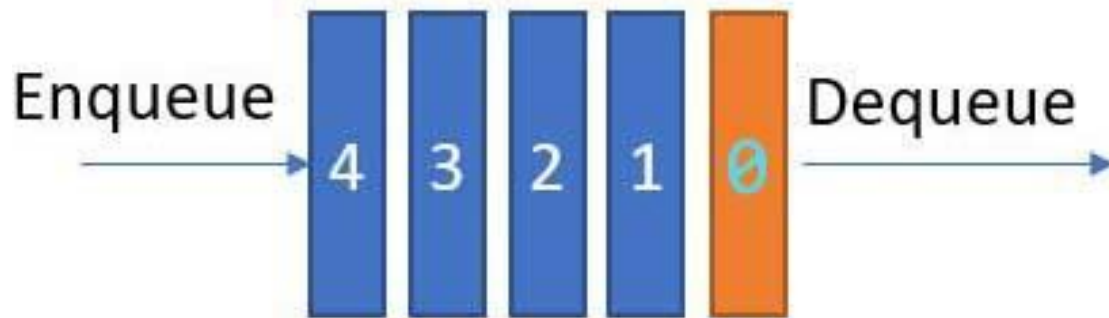
► 使用堆疊物件：

```
Sub Main()  
    Dim s As Stack(Of Integer) = New Stack(Of Integer)()  
    s.Push(5)  
    s.Push(10)  
    s.Push(15)  
    s.Push(20)  
    For Each element As Integer In s  
        Console.WriteLine(element)  
    Next  
    s.Pop()  
    Console.WriteLine()  
    For Each element As Integer In s  
        Console.WriteLine(element)  
    Next  
    Console.Read()  
End Sub
```



佇列(Queue)

- ▶ 顧名思義是一種像排隊一樣的概念
- ▶ 在這個模式下我們可以知道它是一種先進先出(First-In-First-Out, **FIFO**)的排程
- ▶ 存入稱**Enqueue**，取出稱**Dequeue**



佇列(Queue)

- ▶ 佇列可以用陣列實現，也可以用內建的佇列物件
- ▶ 佇列常用的方法如下：

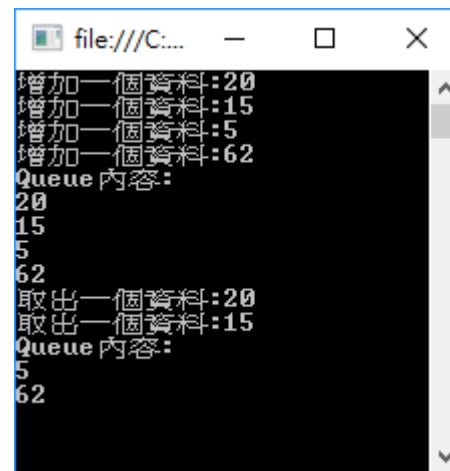
名稱	說 明
Count	取得佇列中目前的項目數量
Dequeue	從佇列前端取出一個項目，同時將其移除
Enqueue	從佇列尾端加入一個項目
Peek	從佇列前端取出一個項目，但不移除

佇列(Queue)

▶ 使用陣列：

```
Dim Q(10), Front, Rear As Integer
Sub Main()
    Front = 0 : Rear = 0
    Enqueue(20) : Enqueue(15) : Enqueue(5) : Enqueue(62)
    PrintQ()
    Dequeue() : Dequeue()
    PrintQ()
    Console.Read()
End Sub
Sub Enqueue(x) '存入資料，往後移動rear指標
    If Rear <= 10 Then
        Q(Rear) = x
        Rear += 1
        Console.WriteLine("增加一個資料:" & x)
    Else
        Console.WriteLine("Queue已經滿了")
    End If
End Sub
Sub Dequeue() '取出資料，往前移動Front指標
    If Front < Rear Then
        Console.WriteLine("取出一個資料:" & Q(Front))
        Front += 1
    Else
        Console.WriteLine("Queue是空的")
    End If
End Sub
```

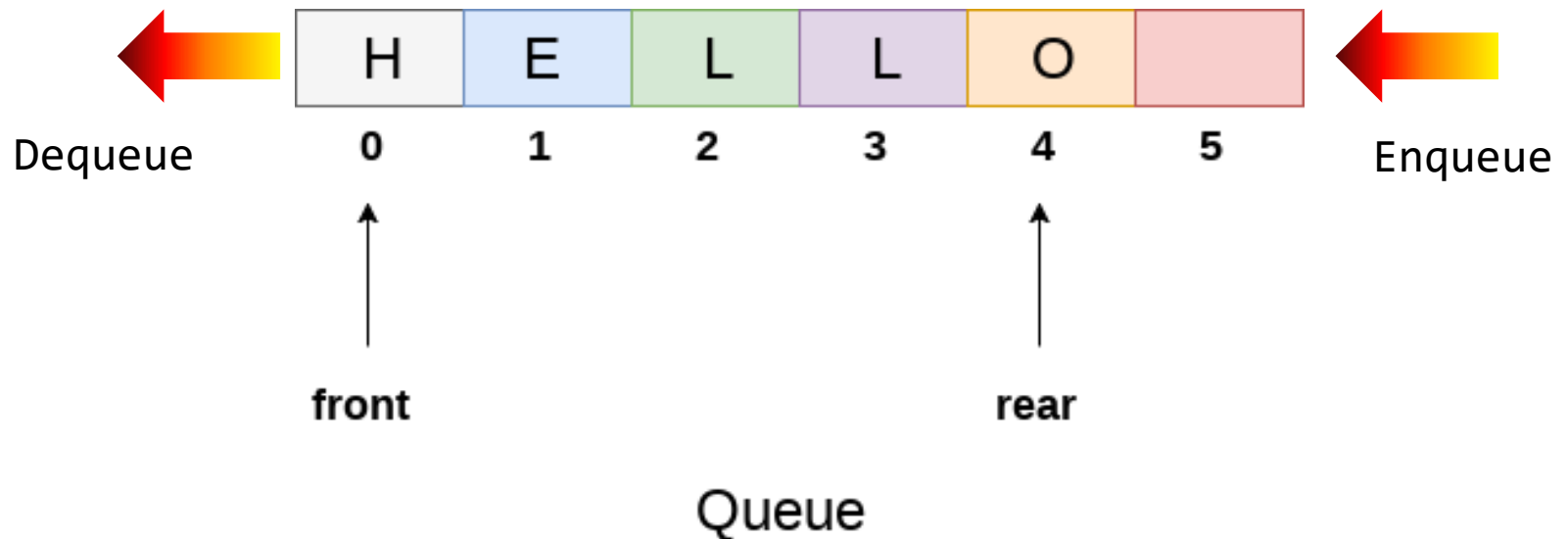
```
Sub PrintQ() '印出佇列內容
    Console.WriteLine("Queue內容:")
    For i = Front To Rear - 1
        Console.WriteLine(Q(i))
    Next
End Sub
```



```
file:///C:...
增加一個資料:20
增加一個資料:15
增加一個資料:5
增加一個資料:62
Queue內容:
20
15
5
62
取出一個資料:20
取出一個資料:15
Queue內容:
5
62
```

佇列(Queue)

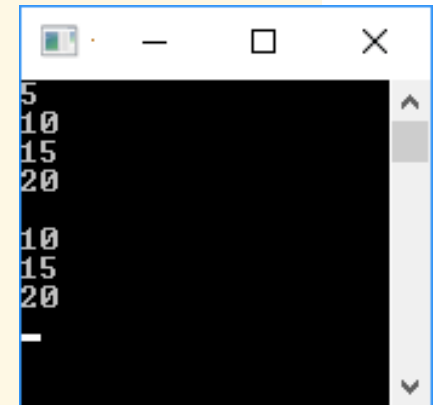
- ▶ 因為動態陣列無法刪除第0個元素，故必須有兩個指標，來標示佇列的頭(Front)和尾(Rear)



佇列(Queue)

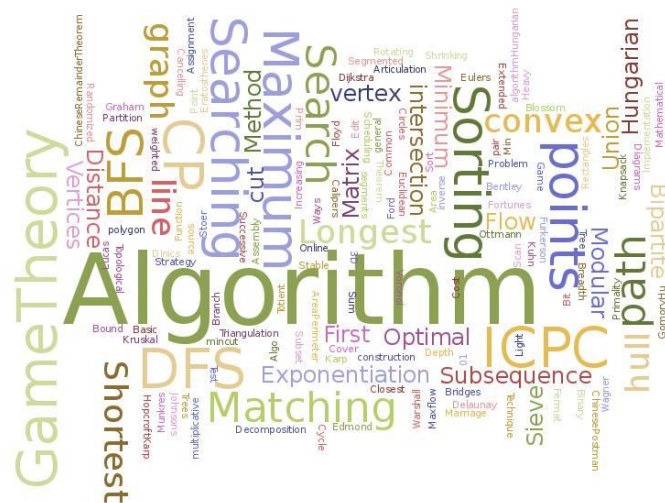
▶ 使用佇列物件：

```
Sub Main()  
    Dim q As Queue(Of Integer) = New Queue(Of Integer)()  
    q.Enqueue(5)  
    q.Enqueue(10)  
    q.Enqueue(15)  
    q.Enqueue(20)  
    For Each element As Integer In q  
        Console.WriteLine(element)  
    Next  
    q.Dequeue()  
    Console.WriteLine()  
    For Each element As Integer In q  
        Console.WriteLine(element)  
    Next  
    Console.Read()  
End Sub
```



說明

- ▶ 物件導向(OOP)是很方便的，多多使用物件
- ▶ 堆疊(Stack)和佇列(Queue)的詳細研究會放在「[資料結構\(Data Structures\)](#)」的課程中說明，有興趣的同學可以再深入研究。
- ▶ 資料結構又和「[演算法\(algorithm\)](#)」有關，是電腦科技人必備的知識



休息一下~



排序和搜尋



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：
 - ▶ 1.氣泡排序法(Bubble Sort)
 - ▶ 2.選擇排序法(Selection Sort)
 - ▶ 3.插入排序法(Insertion Sort)
 - ▶ 4.快速排序法(Quick Sort)
 - ▶ 5.合併排序法(Merge Sort)
- ▶ 不同的排序方法效率不同，應用的地方也不同。



排序

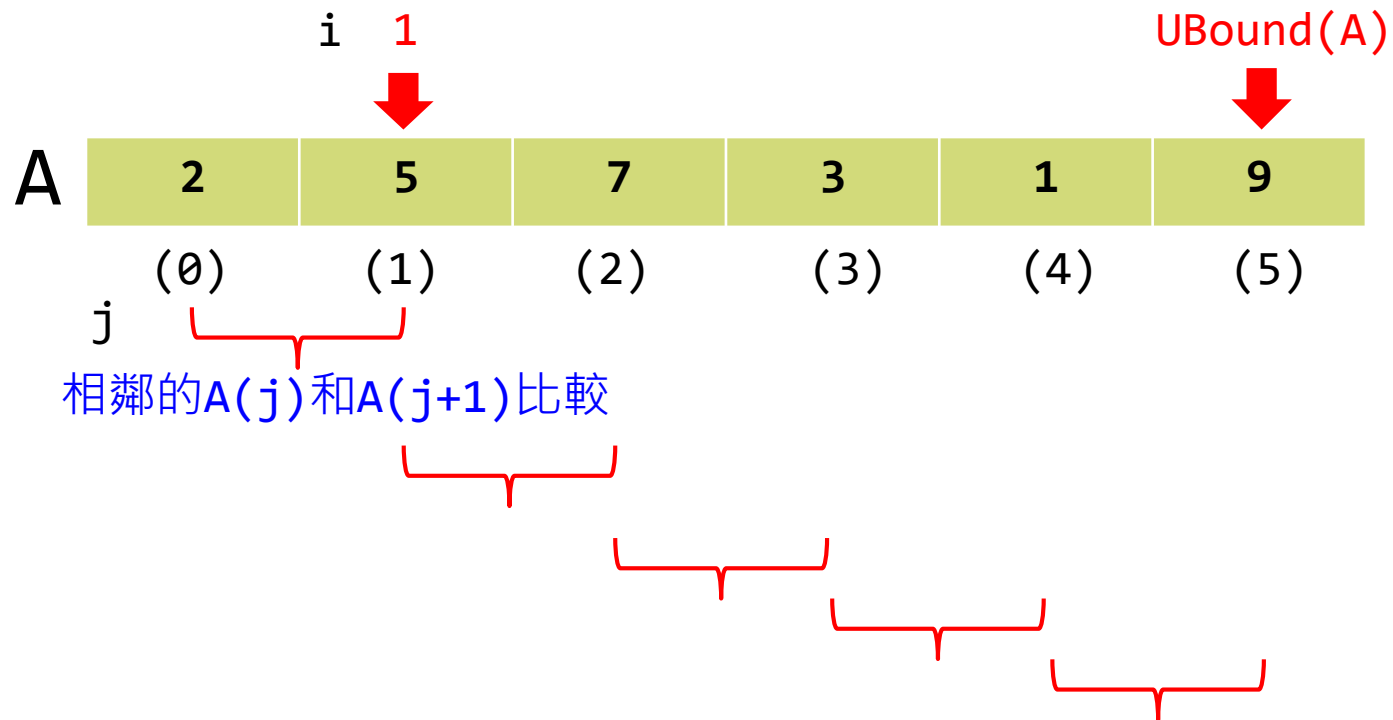
- ▶ 氣泡排序法(Bubble Sort)：

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。

- ▶ 簡單，但效率欠佳，只適用於少量資料

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：

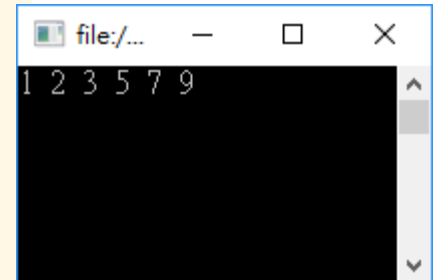
```
For i = 1 To UBound(A)
    For j = 0 To UBound(A) - i
        交換A(j)和A(j+1)
    Next
Next
```

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()
```

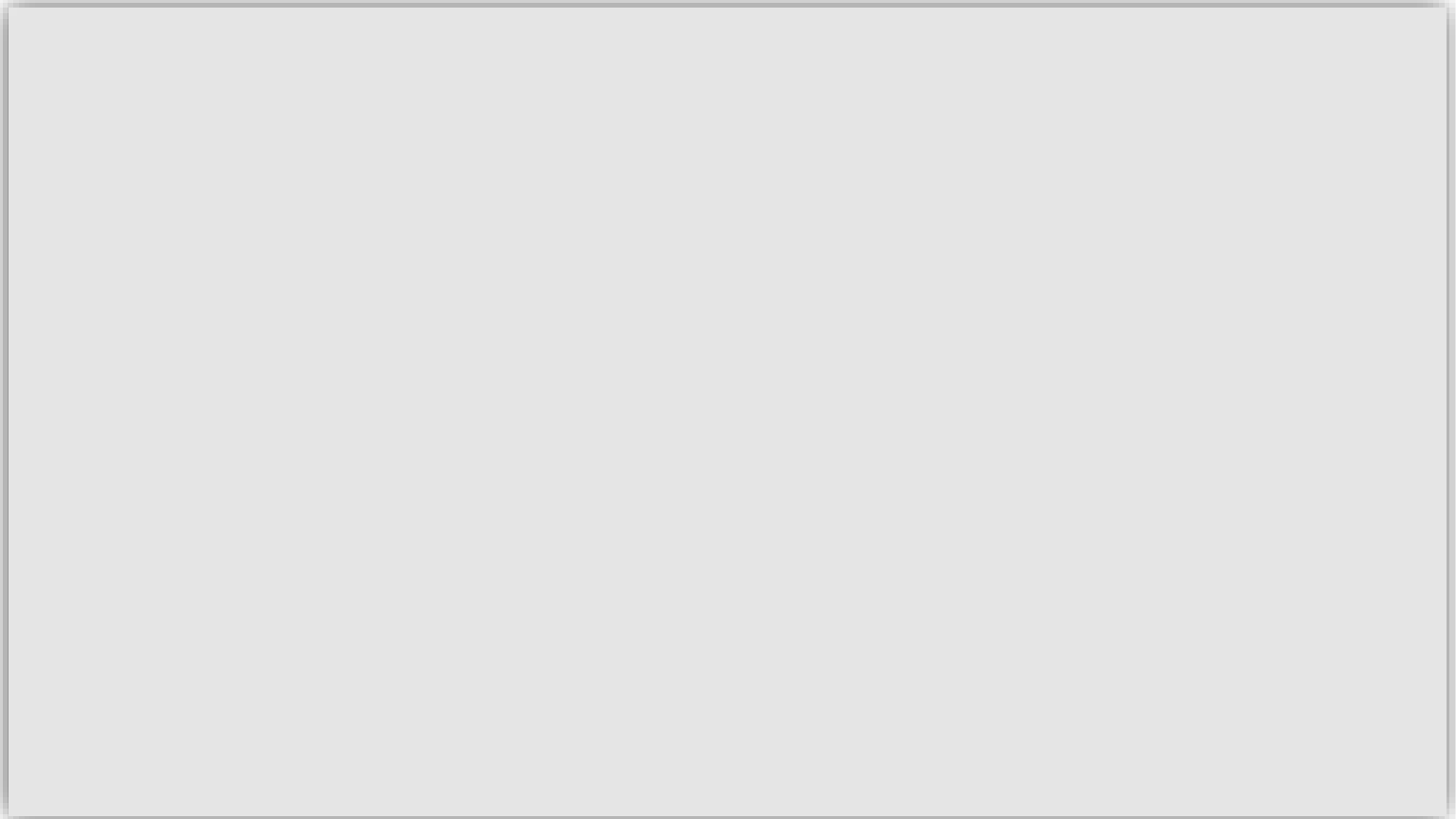
‘交換兩元素

‘印出陣列



氣泡排序法(Bubble Sort)

- ▶ 看看動畫比較容易了解~



氣泡排序法(Bubble Sort)

▶ 參考程式：

```
Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N '輸入數字並存入陣列
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    For i = 1 To N '呼叫排序副程式
        printArray(A) '每一巡迴就印出陣列現況
        bubble(A, i, N) '呼叫氣泡排序
    Next
    printArray(A) '印出陣列最後結果
    Console.Read()
End Sub
```

氣泡排序法(Bubble Sort)

▶ 參考程式：

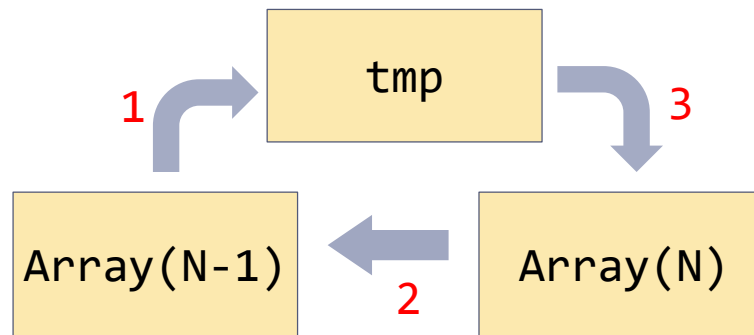
```
Sub bubble(Array, i, N) '氣泡排序一個循環
    Dim tmp As Integer
    While N > i
        If Array(N - 1) > Array(N) Then
            tmp = Array(N - 1)
            Array(N - 1) = Array(N)
            Array(N) = tmp
        End If
        N = N - 1
    End While
End Sub
```

```
Sub printArray(Array) '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

氣泡排序法(Bubble Sort)

- ▶ 兩變數互換需要藉助另一個變數當中介來交換兩變數。

```
If Array(N - 1) > Array(N) Then  
    tmp = Array(N - 1)  
    Array(N - 1) = Array(N)  
    Array(N) = tmp  
End If
```



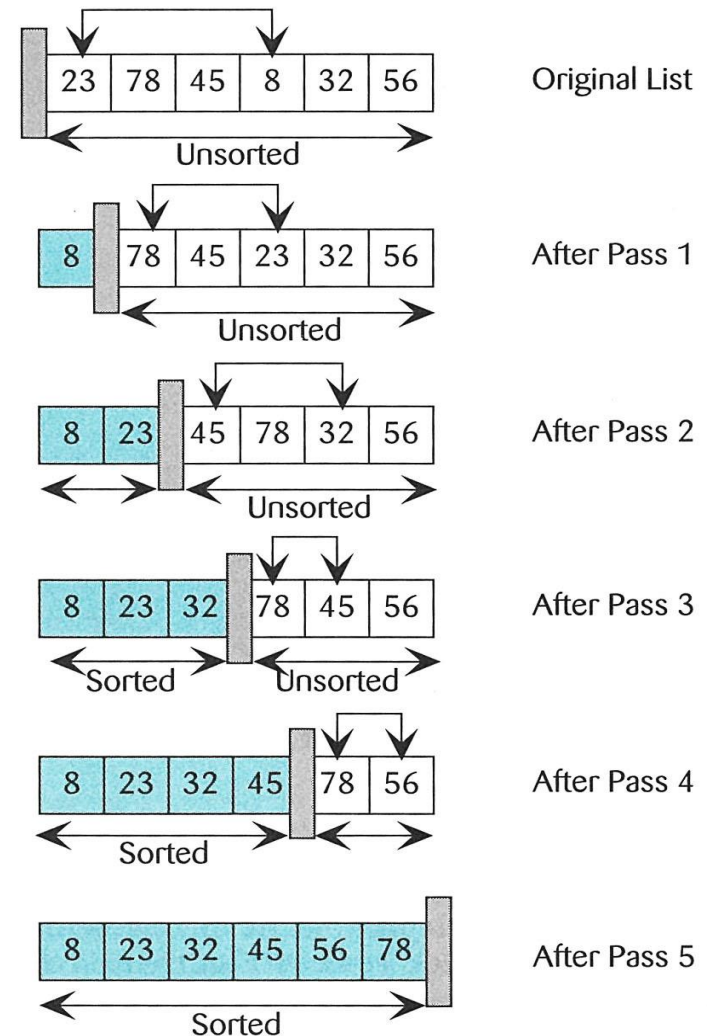
氣泡排序法(Bubble Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Pr...
請輸入陣列大小:6
請輸入第1個數字:9
請輸入第2個數字:4
請輸入第3個數字:8
請輸入第4個數字:2
請輸入第5個數字:7
請輸入第6個數字:3
9      4      8      2      7      3
2      9      4      8      3      7
2      3      9      4      8      7
2      3      4      9      7      8
2      3      4      7      8      9
2      3      4      7      8      9
```

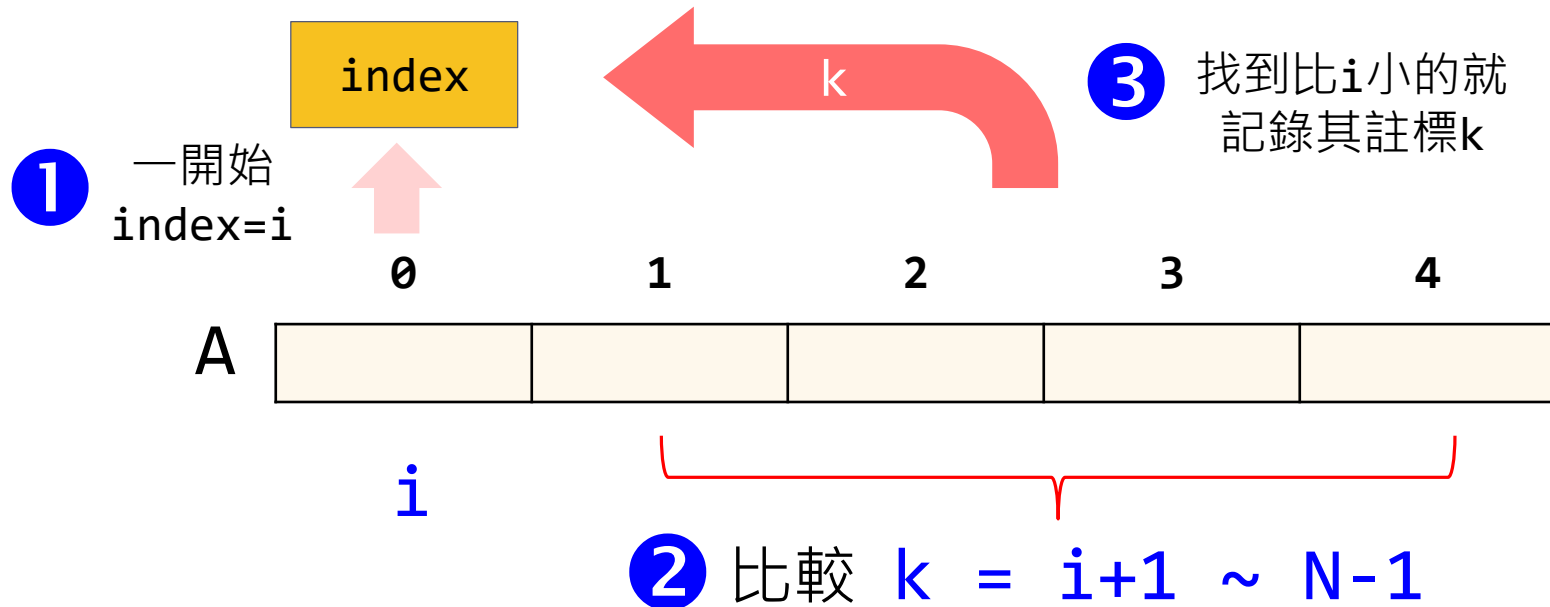
選擇排序法(Selection Sort)

- 先假設第一個元素是最小的，依序向後掃描個元素，若有更小的就記錄起來，最後和第一個元素互換位置，完成一個巡迴，然後第二個元素，依此類推。



選擇排序法(Selection Sort)

► 圖示說明：



4 掃完一遍後將 $A(i)$ 與 $A(k)$ 互換

選擇排序法(Selection Sort)

▶ 參考程式：

```
Sub Main()    '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N    '輸入數字並存入陣列
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    For i = 1 To N    '呼叫排序副程式
        printArray(A)    '每一巡迴就印出陣列現況
        selection(A, i, N)    '呼叫選擇排序
    Next
    printArray(A)    '印出陣列最後結果
    Console.Read()
End Sub
```

選擇排序法(Selection Sort)

▶ 參考程式：

```
Sub selection(Array, i, N)    '選擇排序一個循環
    Dim index, k, tmp As Integer
    index = i                '現在位置
    k = i + 1                '尋找範圍：現在位置之後到陣列結尾
    While k <= N              '尋找比現在元素更小的值，並記錄位置
        If Array(k) < Array(index) Then index = k
        k += 1
    End While
    If index <> i Then        '如果有記錄到更小的值就交換，否則略過
        tmp = Array(i)
        Array(i) = Array(index)
        Array(index) = tmp
    End If
End Sub

Sub printArray(Array)        '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

選擇排序法(Selection Sort)

▶ 執行結果：

請輸入陣列大小:6
請輸入第0個數字:9
請輸入第1個數字:4
請輸入第2個數字:7
請輸入第3個數字:2
請輸入第4個數字:8
請輸入第5個數字:3
[9, 4, 7, 2, 8, 3]

Pass 1 [2, 4, 7, 9, 8, 3]



Pass 2 [2, 3, 7, 9, 8, 4]



Pass 3 [2, 3, 4, 9, 8, 7]



Pass 4 [2, 3, 4, 7, 8, 9]



Pass 5 [2, 3, 4, 7, 8, 9]



result [2, 3, 4, 7, 8, 9]

最後一次無須交換

插入排序法(Insertion Sort)

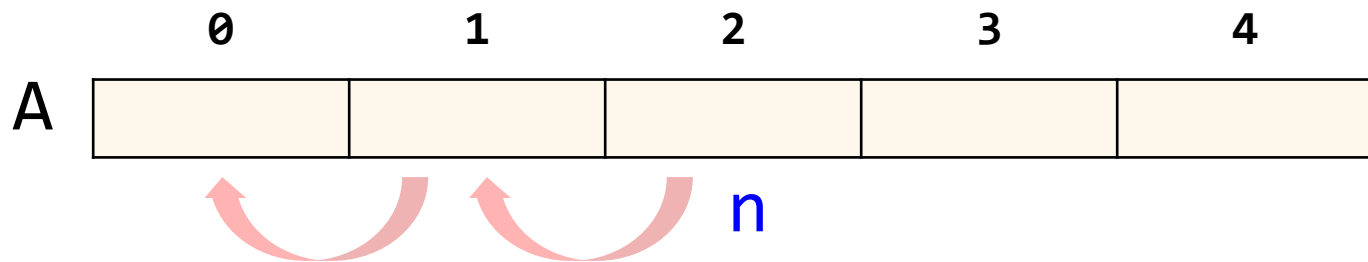
- ▶ 假設由小到大排序，每個元素和其左邊比較，若比左側小則交換，直到比左側大為止。

Insertion Sort							
35	97	19	4	57	27	98	36
0	1	2	3	4	5	6	7



插入排序法(Insertion Sort)

► 交換原則：



比較 $A(n)$ 和 $A(n-1)$

- ❶ if $A(n) < A(n-1) \rightarrow$ 交換
- ❷ 直到 $A(1)$ 或 $A(n) \geq A(n-1) \rightarrow$ 停止
- ❸ 下一個，重複直到陣列最後一個元素

插入排序法(Insertion Sort)

▶ 參考程式：

```
Sub Main()    '主程式
    Dim A(), N, k As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 1 To N    '輸入數字並存入陣列
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    k = 2
    While k <= N
        printArray(A)    '每一巡迴就印出陣列現況
        insert(A, k)    '呼叫選擇排序
        k += 1
    End While
    printArray(A)    '印出陣列最後結果
    Console.Read()
End Sub
```

插入排序法(Insertion Sort)

▶ 參考程式：

```
Sub insert(Array, i) '插入排序法一個循環
    Dim k, tmp As Integer
    k = i
    While k >= 1 '從第i個位置開始向前依序比較
        If Array(k) < Array(k - 1) Then '比左側小則交換兩元素
            tmp = Array(k - 1)
            Array(k - 1) = Array(k)
            Array(k) = tmp
        Else '否則結束本pass
            Exit While
        End If
        k = k - 1
    End While
End Sub

Sub printArray(Array) '印出陣列現況
    For i = 1 To UBound(Array)
        Console.Write(Array(i) & vbTab)
    Next
    Console.WriteLine()
End Sub
```

插入排序法(Insertion Sort)

▶ 執行結果：

請輸入陣列大小:6
請輸入第0個數字:9
請輸入第1個數字:4
請輸入第2個數字:1
請輸入第3個數字:5
請輸入第4個數字:13
請輸入第5個數字:8

Pass 1 [9, 4, 1, 5, 13, 8]


Pass 2 [4, 9, 1, 5, 13, 8]


Pass 3 [1, 4, 9, 5, 13, 8]

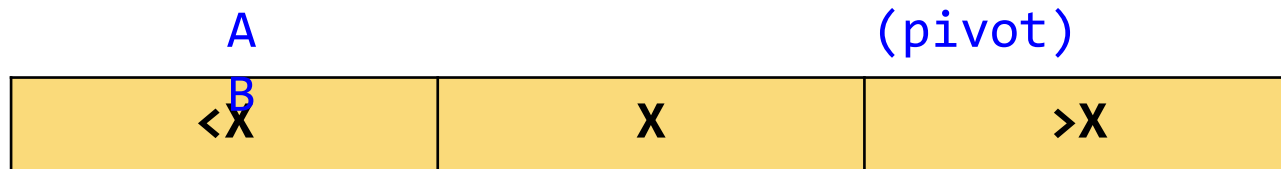

Pass 4 [1, 4, 5, 9, 13, 8] 無須交換


Pass 5 [1, 4, 5, 9, 13, 8]


result [1, 4, 5, 8, 9, 13]

快速排序法(Quick Sort)

- ▶ 快速排序法的原理是選定一個數X(稱「基準」pivot)，然後設法將數列分成三份，如圖：



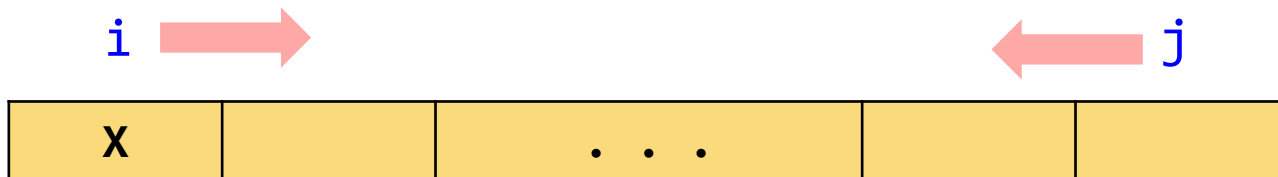
- ▶ 使用遞迴，以一樣的方式再分別處理A段和B段，直到所有數字都處理完畢。
- ▶ 遞迴到最底部時，數列的大小是零或一，也就是已經排序好了。
- ▶ 快速排序通常明顯比其他演算法更快。



快速排序法(Quick Sort)

▶ 演算過程如下：

- ▶ 1. 令數列中最左邊的數為 $X(\text{pivot})$ 。
- ▶ 2. 設立兩個指標 i 和 j 。
- ▶ 3. 將 i 往右移，直到找到第一個 $A(i) \geq X$ 為止。
- ▶ 4. 將 j 往左移，直到找到第一個 $A(j) \leq X$ 為止。
- ▶ 5. 將 $A(i)$ 與 $A(j)$ 互換。
- ▶ 6. 重複以上動作直到 $i \geq j$ 為止。
- ▶ 7. 將 X 與 $A(i)$ 互換。



快速排序法(Quick Sort)

▶ 原理示範說明：

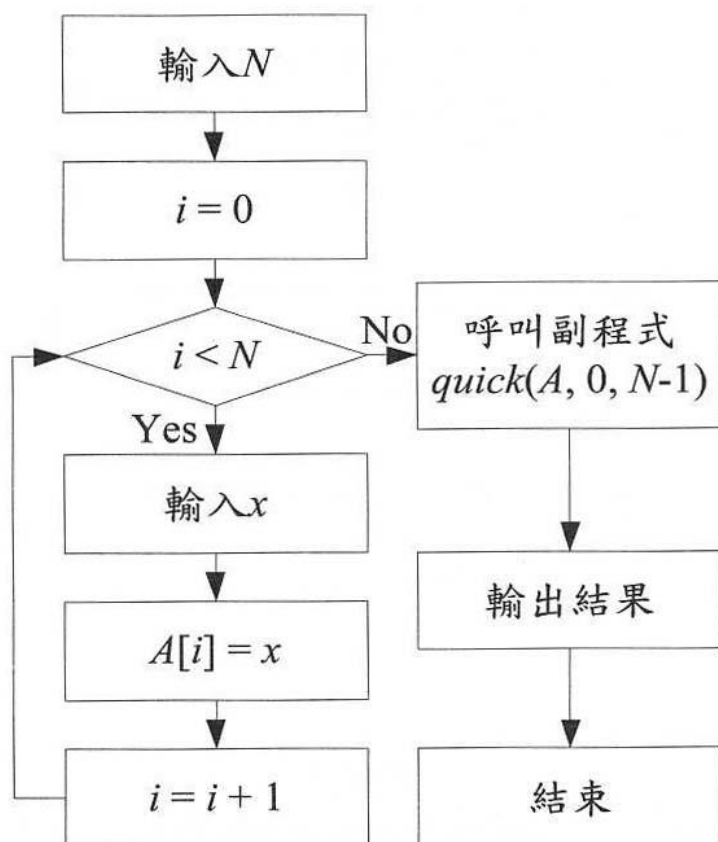
arr



快速排序法(Quick Sort)

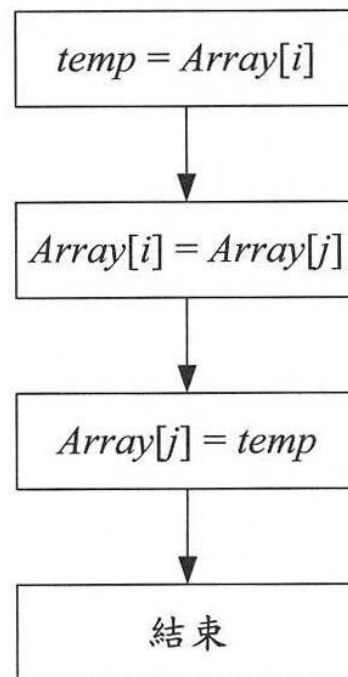
▶ 流程圖：

主程式



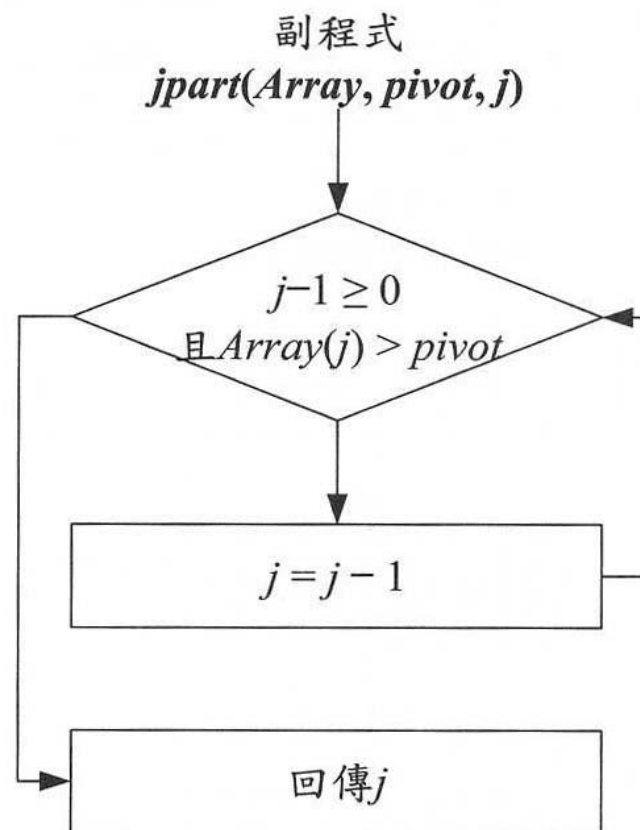
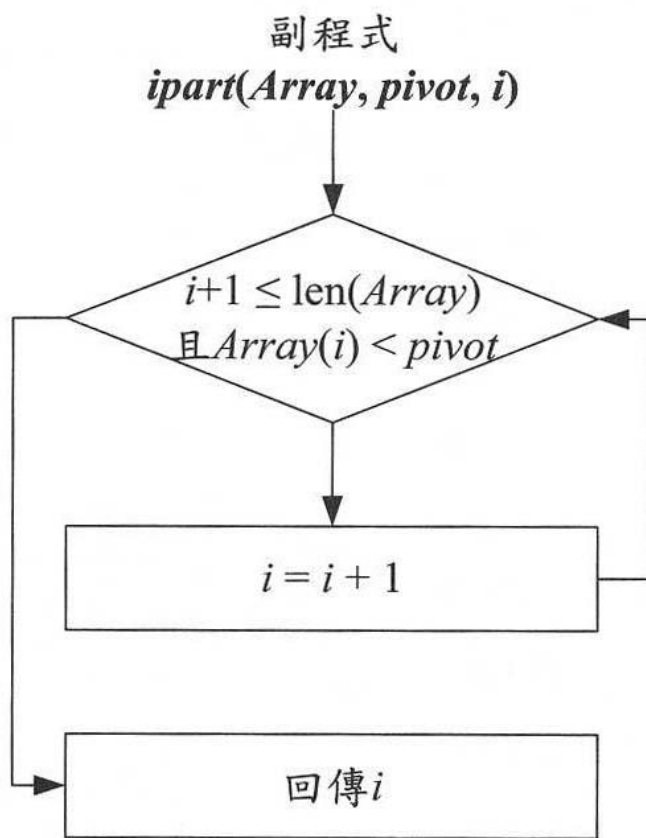
副程式

swap(Array, i, j)



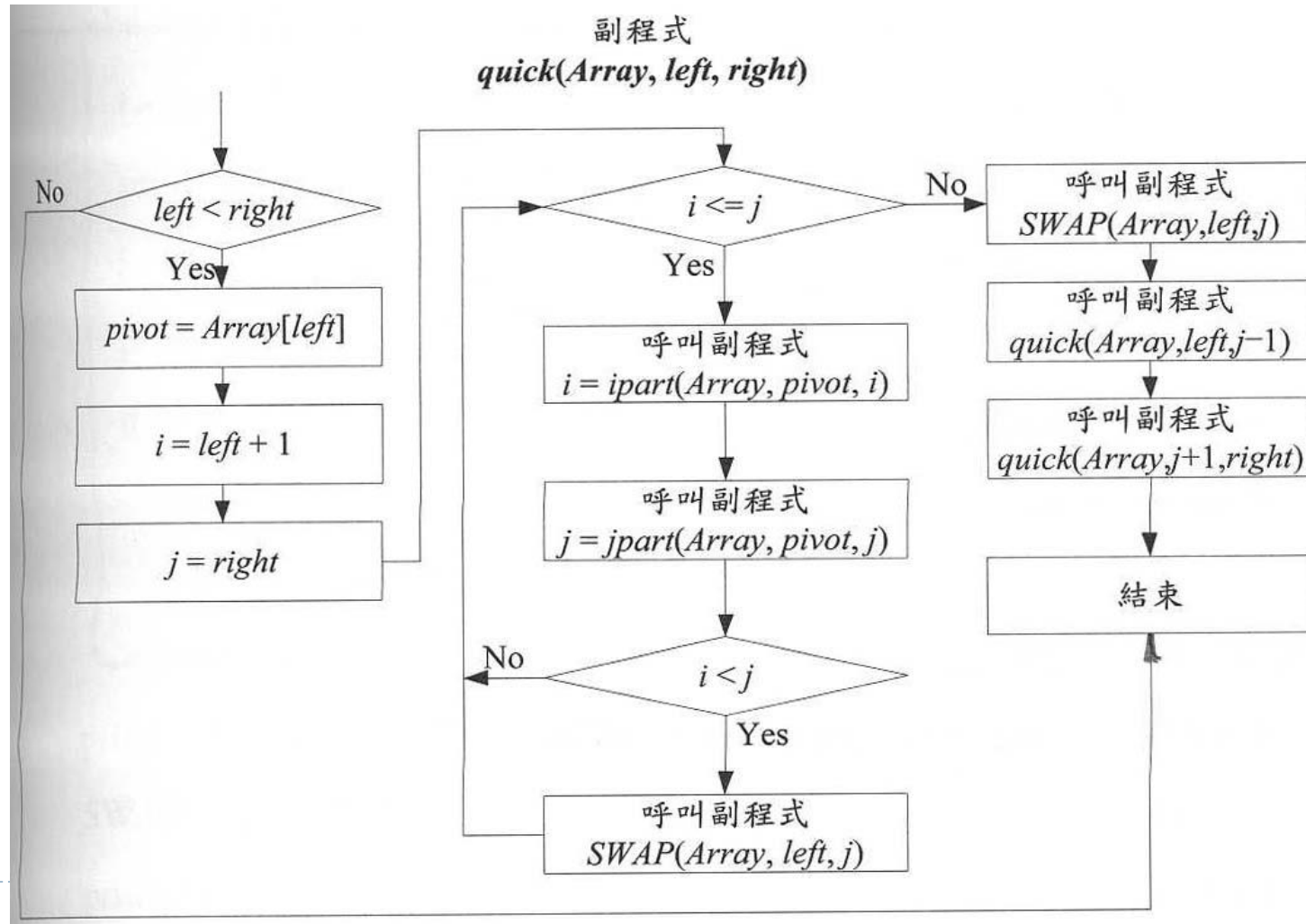
快速排序法(Quick Sort)

▶ 流程圖：



快速排序法(Quick Sort)

▶ 流程圖：



快速排序法(Quick Sort)

▶ 參考程式：`Sub quick(Array, left, right)` '快速排序，程式較冗長是因為要印出執行的過程

```
Dim pivot, i, j As Integer
Console.WriteLine("Q: left= " & left & ",right= " & right)
If left < right Then
    pivot = Array(left)
    Console.WriteLine("pivot = A(" & left & ") = " & pivot)
    i = left + 1
    j = right
    While i <= j
        i = ipart(Array, pivot, i)
        Console.WriteLine("i = " & i)
        j = jpart(Array, pivot, j)
        Console.WriteLine("j = " & j)
        If i < j Then
            Console.WriteLine("i<j，執行交換A(" & i & ")和A(" & j & ")")
            SWAP(Array, i, j)
            printArray(Array)
        End If
    End While
    Console.WriteLine("交換pivot和A(" & j & ")")
    SWAP(Array, left, j)
    printArray(Array)
    quick(Array, left, j - 1)
    quick(Array, j + 1, right)
End If
End Sub
```

快速排序法(Quick Sort)

▶ 參考程式：

'交換兩元素

```
Sub SWAP(Array, i, j)
    Dim temp As Integer
    temp = Array(i)
    Array(i) = Array(j)
    Array(j) = temp
```

End Sub

'將i往右移，直到找到第一個A(i)>=X為止

```
Function ipart(Array, pivot, i)
    While i + 1 <= UBound(Array) And Array(i) < pivot
        i += 1
    End While
    Return i
```

End Function

'將j往左移，直到找到第一個A[j]<=X為止

```
Function jpart(Array, pivot, j)
    While j - 1 >= 0 And Array(j) > pivot
        j -= 1
    End While
    Return j
```

End Function

快速排序法(Quick Sort)

▶ 參考程式：

```
Sub printArray(A) '印出陣列現況
    For i = 0 To UBound(A) - 1
        Console.Write(A(i) & vbTab)
    Next
    Console.WriteLine()
End Sub

Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1 '輸入陣列資料
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    printArray(A) '印出原陣列
    quick(A, 0, N - 1) '呼叫快速排序
    printArray(A) '印出排序結果
    Console.Read()
End Sub
```

快速排序法(Quick Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects/ConsoleApplication4/bin/Debug/ConsoleA...
請輸入陣列大小:10
請輸入第0個數字:11
請輸入第1個數字:20
請輸入第2個數字:3
請輸入第3個數字:15
請輸入第4個數字:16
請輸入第5個數字:24
請輸入第6個數字:9
請輸入第7個數字:31
請輸入第8個數字:25
請輸入第9個數字:28
11    20    3    15    16    24    9    31    25    28
Q: left= 0,right= 9
pivot = A(0) = 11
i = 1
j = 6
i<j, 執行交換A(1)和A(6)
11    9    3    15    16    24    20    31    25    28
i = 3
j = 2
交換pivot和A(2)
3    9    11    15    16    24    20    31    25    28
Q: left= 0,right= 1
pivot = A(0) = 3
i = 1
j = 0
交換pivot和A(0)
3    9    11    15    16    24    20    31    25    28
Q: left= 0,right= -1
Q: left= 1,right= 1
Q: left= 3,right= 9
pivot = A(3) = 15
i = 4
j = 3
交換pivot和A(3)
3    9    11    15    16    24    20    31    25    28
```

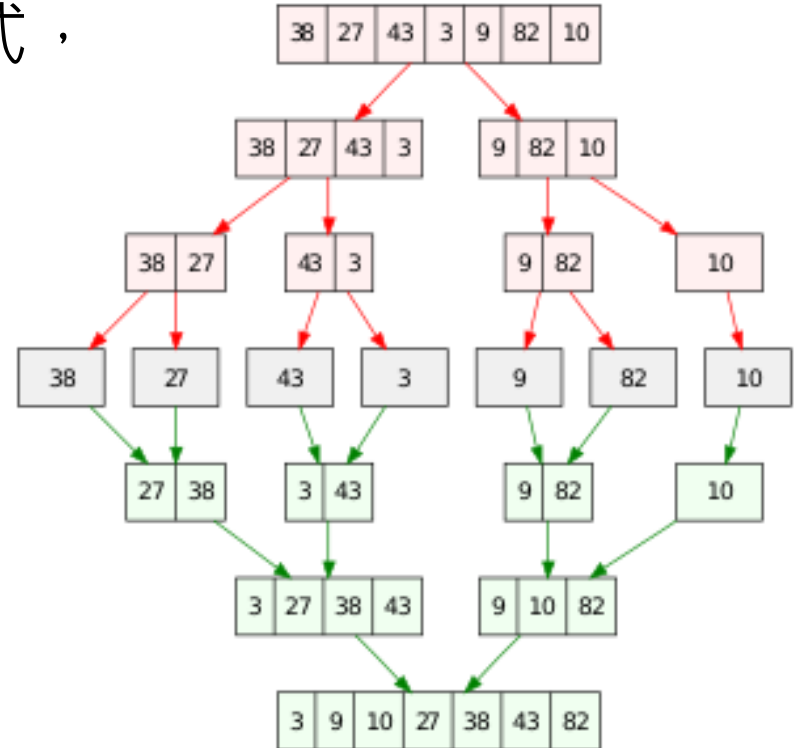
快速排序法(Quick Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects/ConsoleApplication4/bin/Debug/ConsoleA...
交換pivot和A(3)
3    9    11    15    16    24    20    31    25    28
Q: left= 3,right= 2
Q: left= 4,right= 9
pivot = A(4) = 16
i = 5
j = 4
交換pivot和A(4)
3    9    11    15    16    24    20    31    25    28
Q: left= 4,right= 3
Q: left= 5,right= 9
pivot = A(5) = 24
i = 7
j = 6
交換pivot和A(6)
3    9    11    15    16    20    24    31    25    28
Q: left= 5,right= 5
Q: left= 7,right= 9
pivot = A(7) = 31
i = 10
j = 9
交換pivot和A(9)
3    9    11    15    16    20    24    28    25    31
Q: left= 7,right= 8
pivot = A(7) = 28
i = 9
j = 8
交換pivot和A(8)
3    9    11    15    16    20    24    25    28    31
Q: left= 7,right= 7
Q: left= 9,right= 8
Q: left= 10,right= 9
3    9    11    15    16    20    24    25    28    31
```

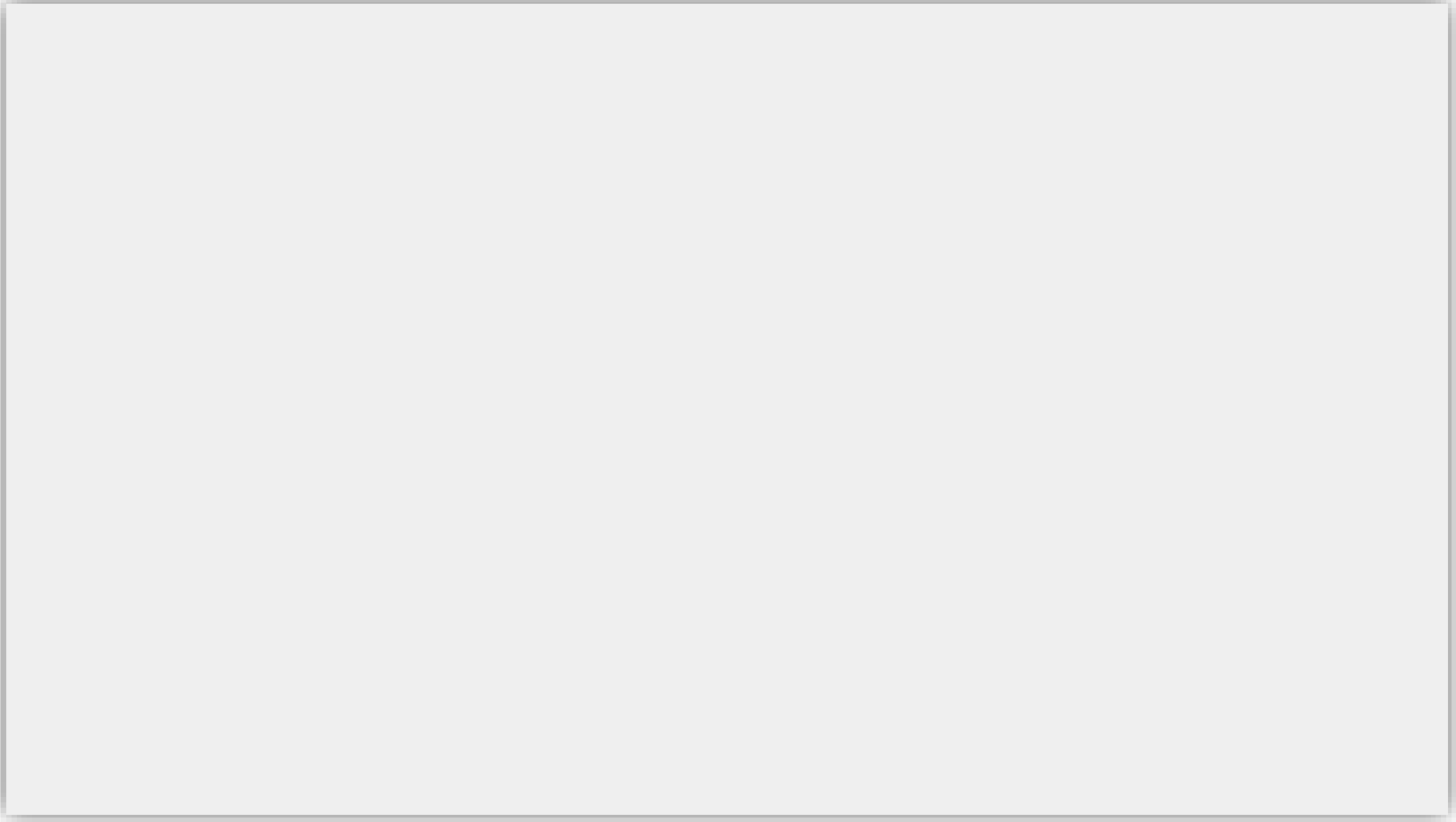
合併排序法(Merge Sort)

- ▶ 合併排序法是依據“合併”來排序的，假設我們有兩個已經排好的數列，我們就可以很容易的將兩個數列合併成一個排序好的數列。
- ▶ 採用先打散，在合併的方式，稱為divide and conquer。
- ▶ 一般都以遞迴來實現。



合併排序法(Merge Sort)

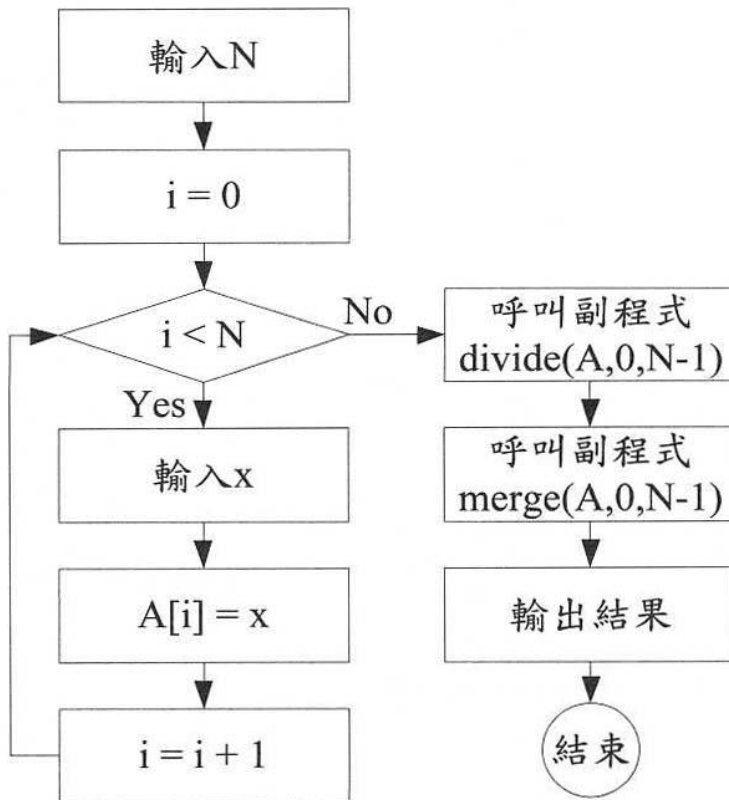
▶ 原理示範說明：



合併排序法(Merge Sort)

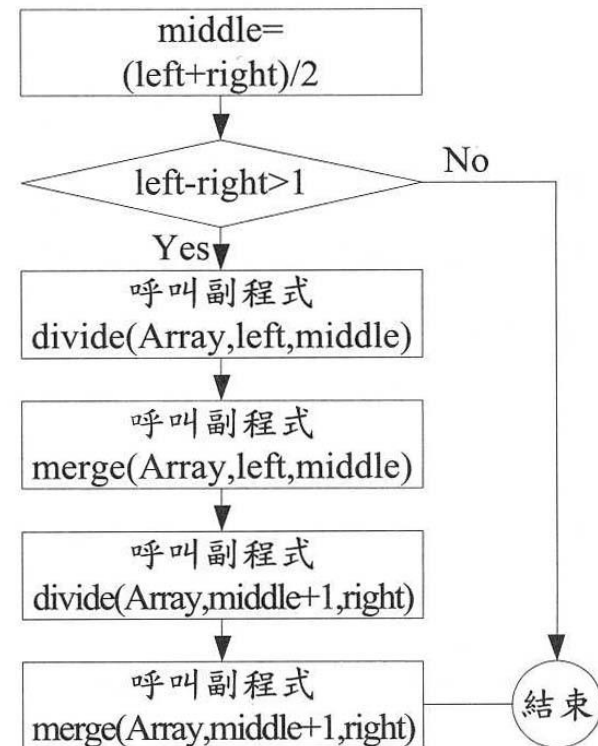
▶ 流程圖：

主程式



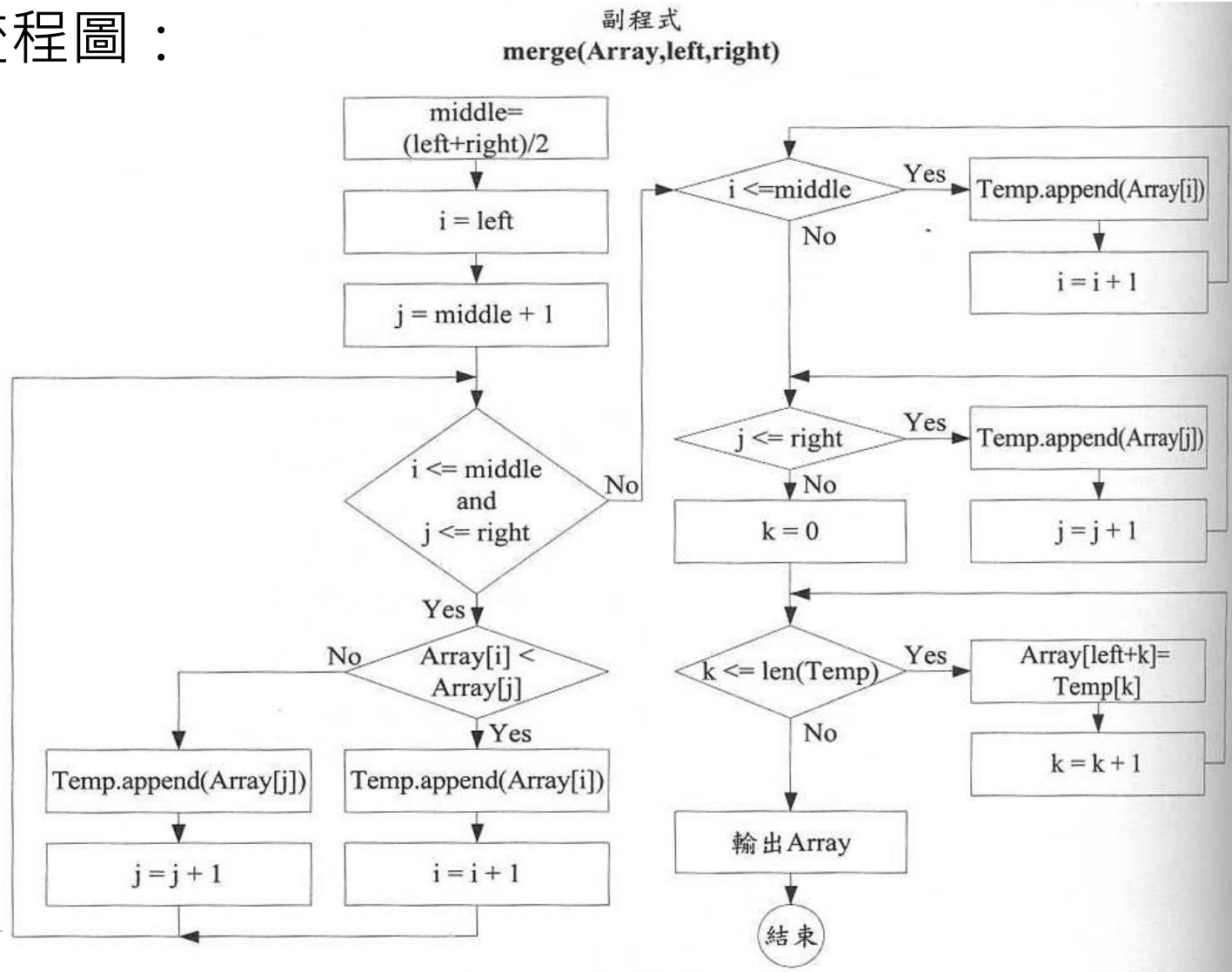
副程式

divide(Array,left,right)



合併排序法(Merge Sort)

▶ 流程圖：



合併排序法(Merge Sort)

▶ 參考程式：

```
Sub divide(Array, left, right) '分割副程式
    Dim middle, leftRange As Integer
    Console.WriteLine("D: left= " & left & ", right= " & right)
    middle = Int((left + right) / 2)
    If (right - left) > 1 Then
        divide(Array, left, middle)
        merge(Array, left, middle)
        divide(Array, middle + 1, right)
        merge(Array, middle + 1, right)
        '印出合併的提示訊息，左半邊要與右半邊合併
        Console.Write("接著將左邊(")
        leftRange = left
        While leftRange <= middle
            Console.Write(Array(leftRange))
            leftRange += 1
        End While
        Console.Write(")及右邊(")
        leftRange = middle + 1
        While leftRange <= right
            Console.Write(Array(leftRange))
            leftRange += 1
        End While
        Console.WriteLine(")合併...")
    End If
End Sub
```

合併排序法

▶ 參考程式：

```
Sub merge(Array, left, right) '合併副程式
    Dim Temp As New ArrayList '此處使用了ArrayList物件
    Dim middle, i, j, k, leftRange As Integer
    Console.WriteLine("M: left= " & left & ", right= " & right)
    middle = Int((left + right) / 2)
    i = left : j = middle + 1
    Console.WriteLine("i= " & i & " j= " & j)
    While i <= middle And j <= right
        If Array(i) < Array(j) Then
            Temp.Add(Array(i)) : i += 1
        Else
            Temp.Add(Array(j)) : j += 1
        End If
    End While
    printTemp(Temp)
    While i <= middle
        Temp.Add(Array(i)) : i += 1
    End While
    While j <= right
        Temp.Add(Array(j)) : j += 1
    End While
    For k = 0 To Temp.Count - 1
        Array(left + k) = Temp(k)
    Next
    printArray(Array) '將排序好的內容印出
    Console.Write("(")
    leftRange = left
    While leftRange <= right
        Console.Write(Array(leftRange)) : leftRange += 1
    End While
    Console.WriteLine(")排序好" & vbCrLf)
End Sub
```

合併排序法(Merge Sort)

▶ 參考程式：

```
Sub printArray(A) '印出排序陣列現況
    For i = 0 To UBound(A) - 1
        Console.Write(A(i) & " ")
    Next
End Sub

Sub printTemp(A) '印出Temp陣列現況
    Console.Write("Temp= (")
    For Each n In A
        Console.Write(n & " ")
    Next
    Console.WriteLine(")")
End Sub
```



合併排序法(Merge Sort)

▶ 參考程式：

```
Sub Main() '主程式
    Dim A(), N As Integer
    Console.Write("請輸入陣列大小:")
    N = Console.ReadLine
    ReDim A(N)
    For i = 0 To N - 1 '輸入陣列資料
        Console.Write("請輸入第" & i & "個數字:")
        A(i) = Console.ReadLine
    Next
    Console.Write("排序前： ")
    printArray(A) '印出原陣列
    Console.WriteLine()
    divide(A, 0, N - 1) '呼叫分割
    merge(A, 0, N - 1) '呼叫合併
    Console.Write("排序完成： ")
    printArray(A) '印出排序結果
    Console.Read()
End Sub
```

合併排序法(Merge Sort)

▶ 執行結果：

```
file:///C:/Users/Orion Liu...
請輸入陣列大小:7
請輸入第0個數字:6
請輸入第1個數字:3
請輸入第2個數字:9
請輸入第3個數字:5
請輸入第4個數字:2
請輸入第5個數字:7
請輸入第6個數字:1
排序前: 6 3 9 5 2 7 1
D: left= 0, right= 6
D: left= 0, right= 3
D: left= 0, right= 1
M: left= 0, right= 1
i= 0 j= 1
Temp= (3 )
3 6 9 5 2 7 1 (36)排序好

D: left= 2, right= 3
M: left= 2, right= 3
i= 2 j= 3
Temp= (5 )
3 6 5 9 2 7 1 (59)排序好

接著將左邊(36)及右邊(59)合併...
M: left= 0, right= 3
i= 0 j= 2
Temp= (3 5 6 )
3 5 6 9 2 7 1 (3569)排序好
```

```
選取 file:///C:/Users/Orio...
D: left= 4, right= 6
D: left= 4, right= 5
M: left= 4, right= 5
i= 4 j= 5
Temp= (2 )
3 5 6 9 2 7 1 (27)排序好

D: left= 6, right= 6
M: left= 6, right= 6
i= 6 j= 7
Temp= ( )
3 5 6 9 2 7 1 (1)排序好

接著將左邊(27)及右邊(1)合併...
M: left= 4, right= 6
i= 4 j= 6
Temp= (1 )
3 5 6 9 1 2 7 (127)排序好

接著將左邊(3569)及右邊(127)合併...
M: left= 0, right= 6
i= 0 j= 4
Temp= (1 2 3 5 6 7 )
1 2 3 5 6 7 9 (1235679)排序好

排序完成: 1 2 3 5 6 7 9
```

休息一下~

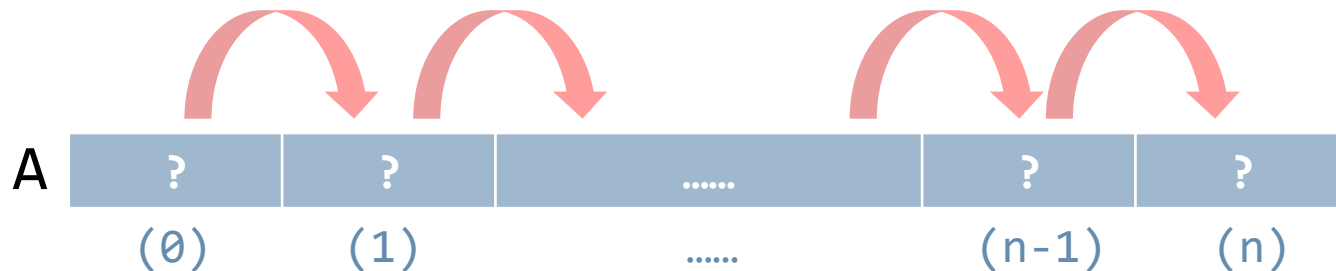


搜尋(Search)

- ▶ 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

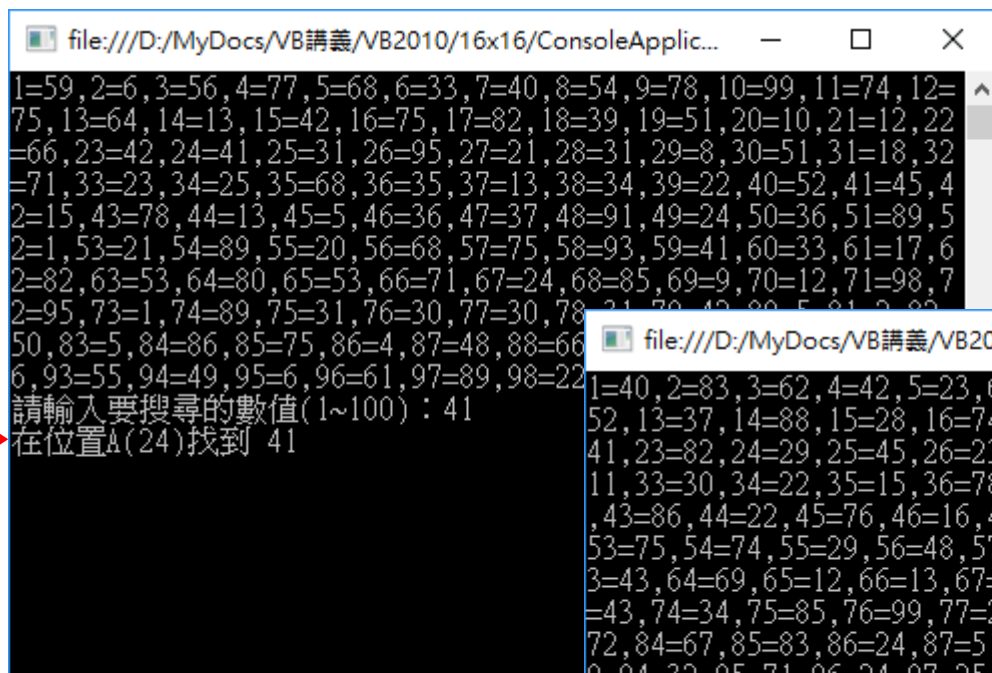
循序搜尋

- ▶ 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=31,79=42,80=5,81=2,82
50,83=5,84=86,85=75,86=4,87=48,88=66
6,93=55,94=49,95=6,96=61,97=89,98=22
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：

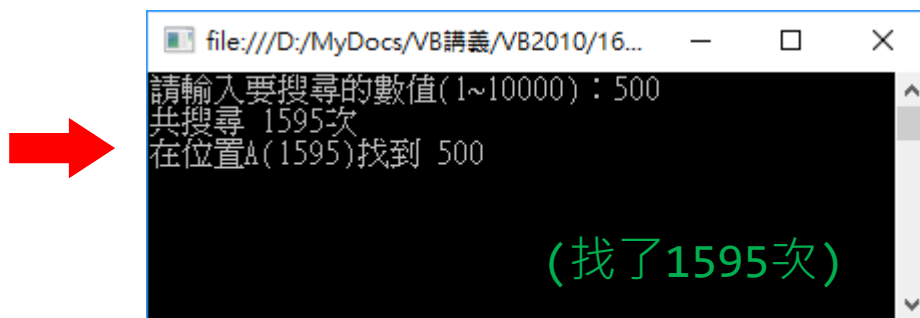
```
Dim A(100), x As Integer
Dim find As Boolean
Randomize()
For i = 1 To 100    '產生100個亂數
    A(i) = Int(Rnd() * 100) + 1
    Console.Write(i & "=" & A(i) & ",")
Next
Console.WriteLine()
Console.Write("請輸入要搜尋的數值(1~100) : ")
x = Console.ReadLine()    '請求輸入
```

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()
```

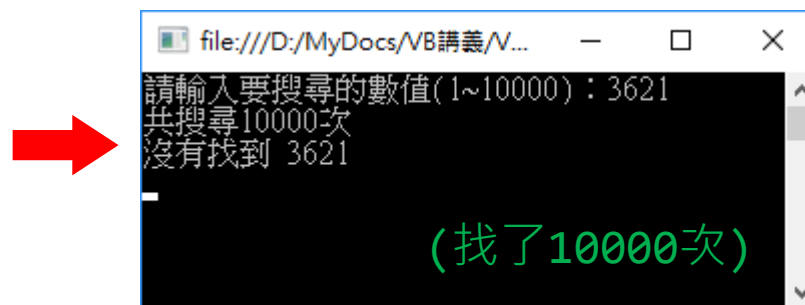
循序搜尋練習

- ▶ 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

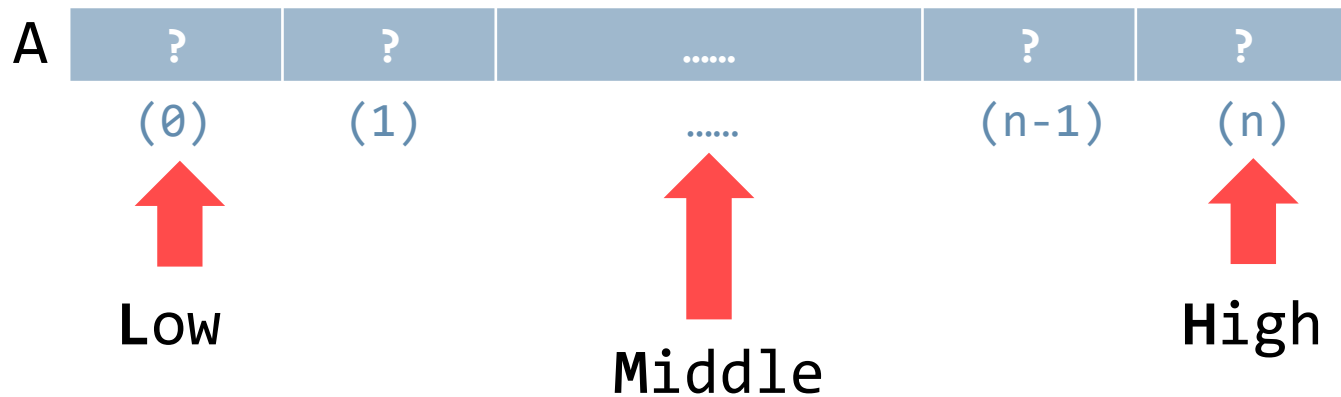
```
Dim A(10000), x As Integer
Dim find As Boolean
Randomize()
For i = 1 To 10000    '產生亂數
    A(i) = Int(Rnd() * 10000) + 1
Next
Console.WriteLine("請輸入要搜尋的數值(1~10000) : ")
x = Console.ReadLine()
find = False
For i = 1 To 10000    '從第一個元素開始依序向後搜尋
    If A(i) = x Then
        Console.WriteLine("共搜尋 " & 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

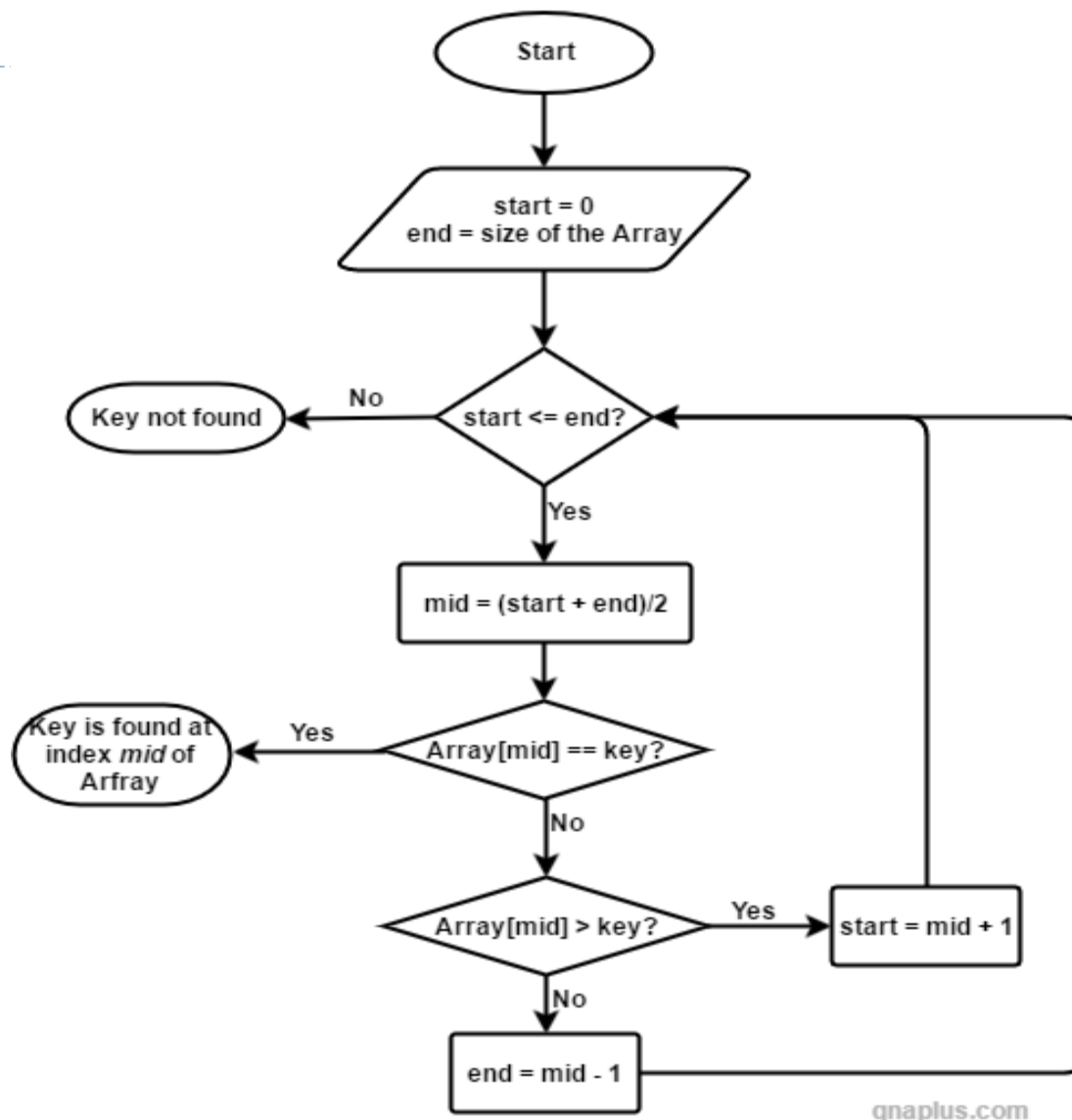


二分搜尋

- ▶ 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

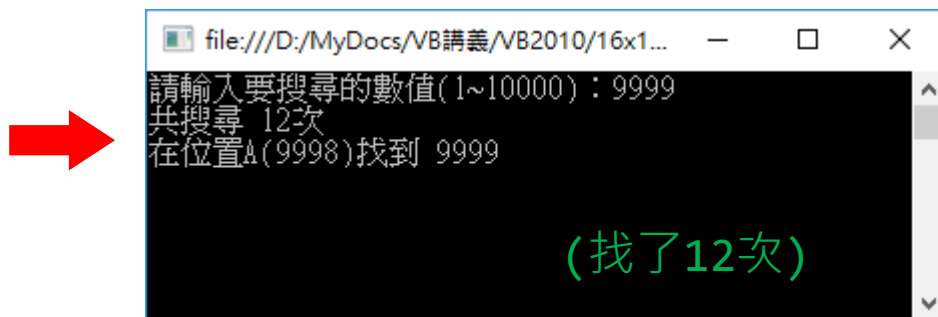
Binary search algorithm: find *key* in a sorted *Array*



qnaplus.com

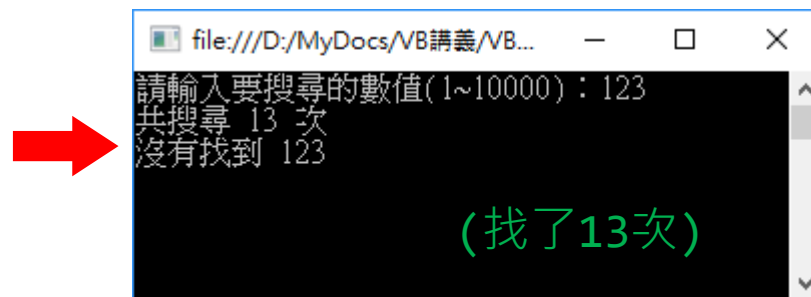
二分搜尋練習

- ▶ 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.WriteLine("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

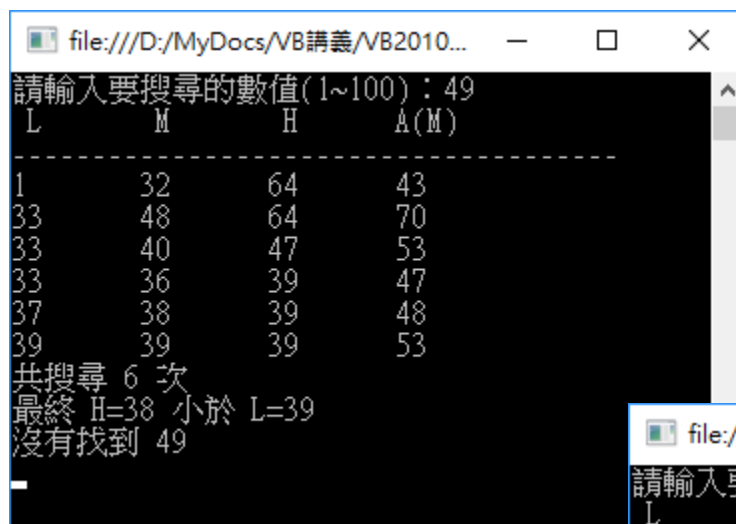
- ▶ 說明：

- ▶ 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

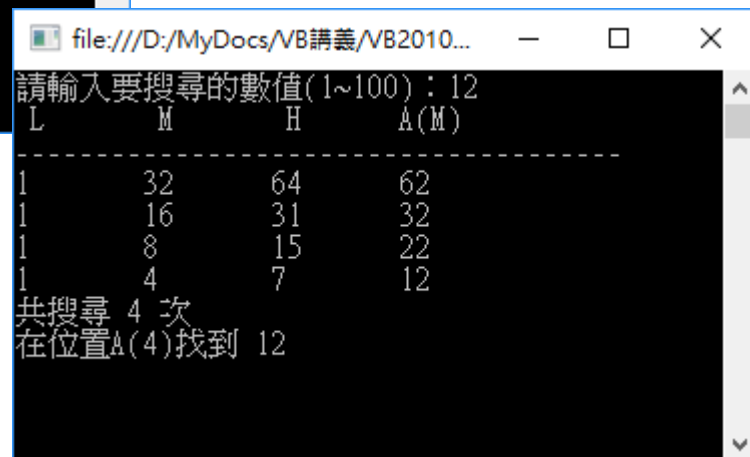
- ▶ 若Log不好算，換一個說法： 2^X 必須 ≥ 64 ，故 $X=6$

二分搜尋練習

- 寫一程式，產生64個1~100的亂數，排序後以二分搜尋尋找指定的值，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
```

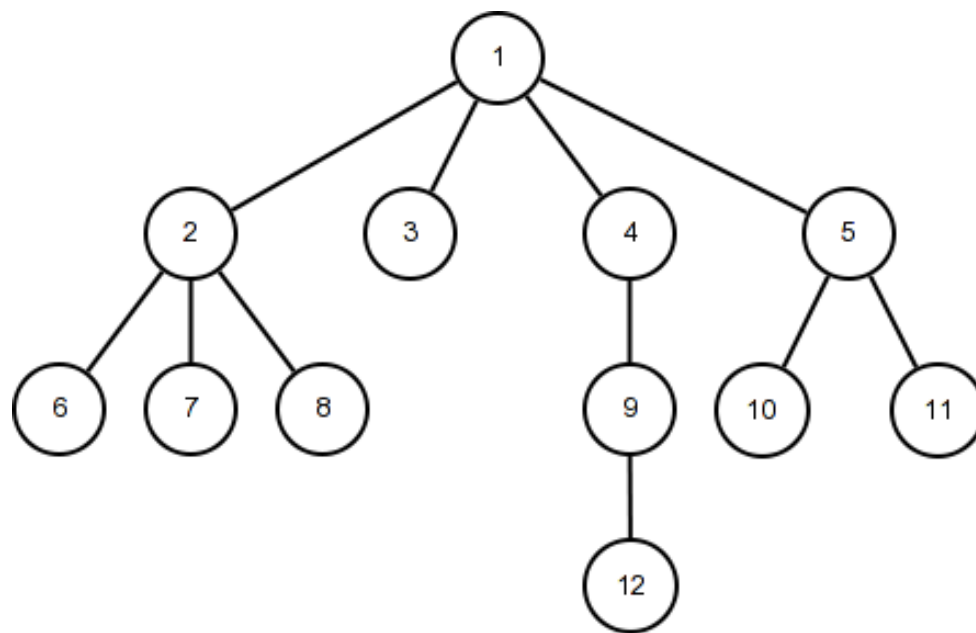
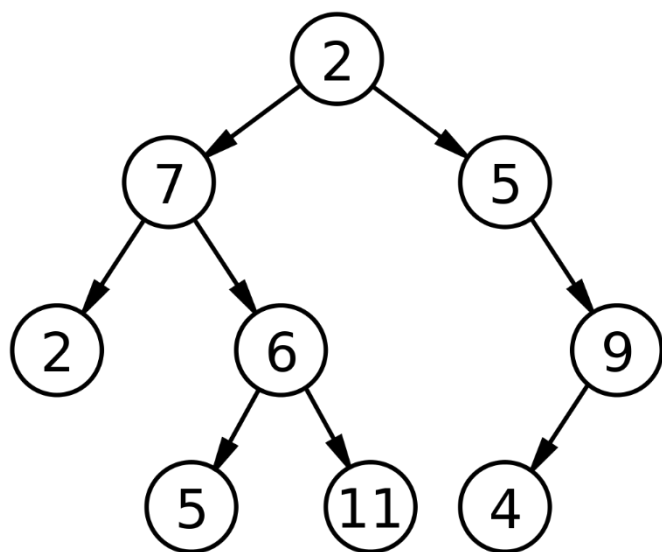
二分搜尋練習



```
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()
```

二元搜尋樹(Binary Search Tree)

- ▶ 樹是一種資料結構方式，使用連結串列(Link List)來組成資料，因為連接的方式像一棵樹，故稱之。
- ▶ 這都是「樹」：



二元搜尋樹(Binary Search Tree)

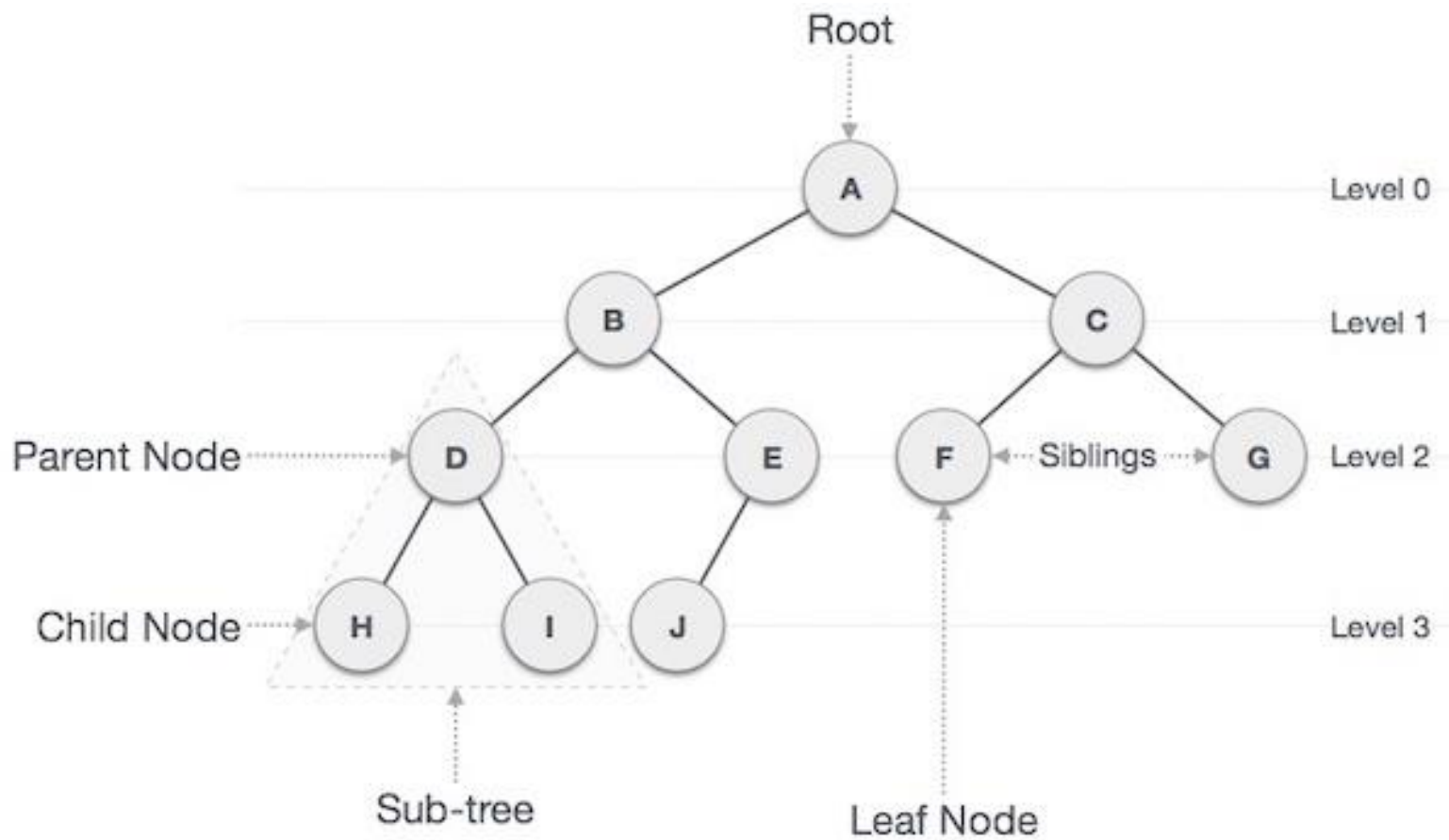
▶ 樹的相關定義：

- ▶ 1.一個樹包含了一組有限的元素，稱為節點(Node)。
- ▶ 2.一組有限的方向線段，稱分支(Branch)，連接到結點上。
- ▶ 3.和這個節點有關的分支數目稱為此節點的degree。
- ▶ 4.當一個分支指向一個節點時，稱此分支為indegree。
- ▶ 5.當一個分支離開一個節點時，稱此分支為outdegree。
- ▶ 6.indegree和outdegree的總和等於該節點的degree。
- ▶ 7.第一個節點稱為根(Root)，除了根節點外，任何節點都必須至少有一個indegree，但outdegree則沒有限制。



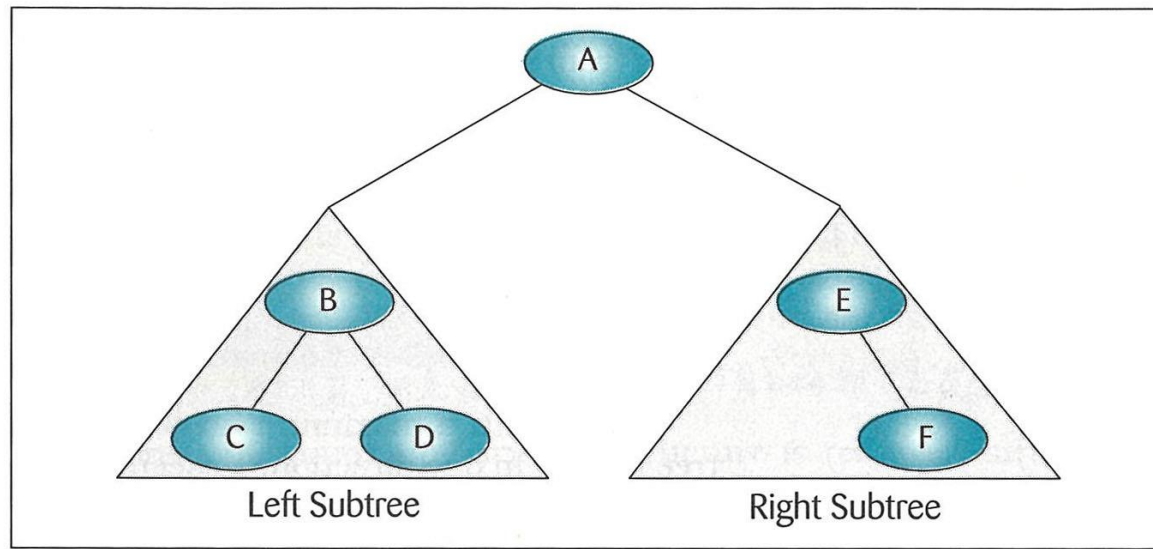
二元搜尋樹(Binary Search Tree)

- ▶ 樹的一些專有名詞：



二元搜尋樹(Binary Search Tree)

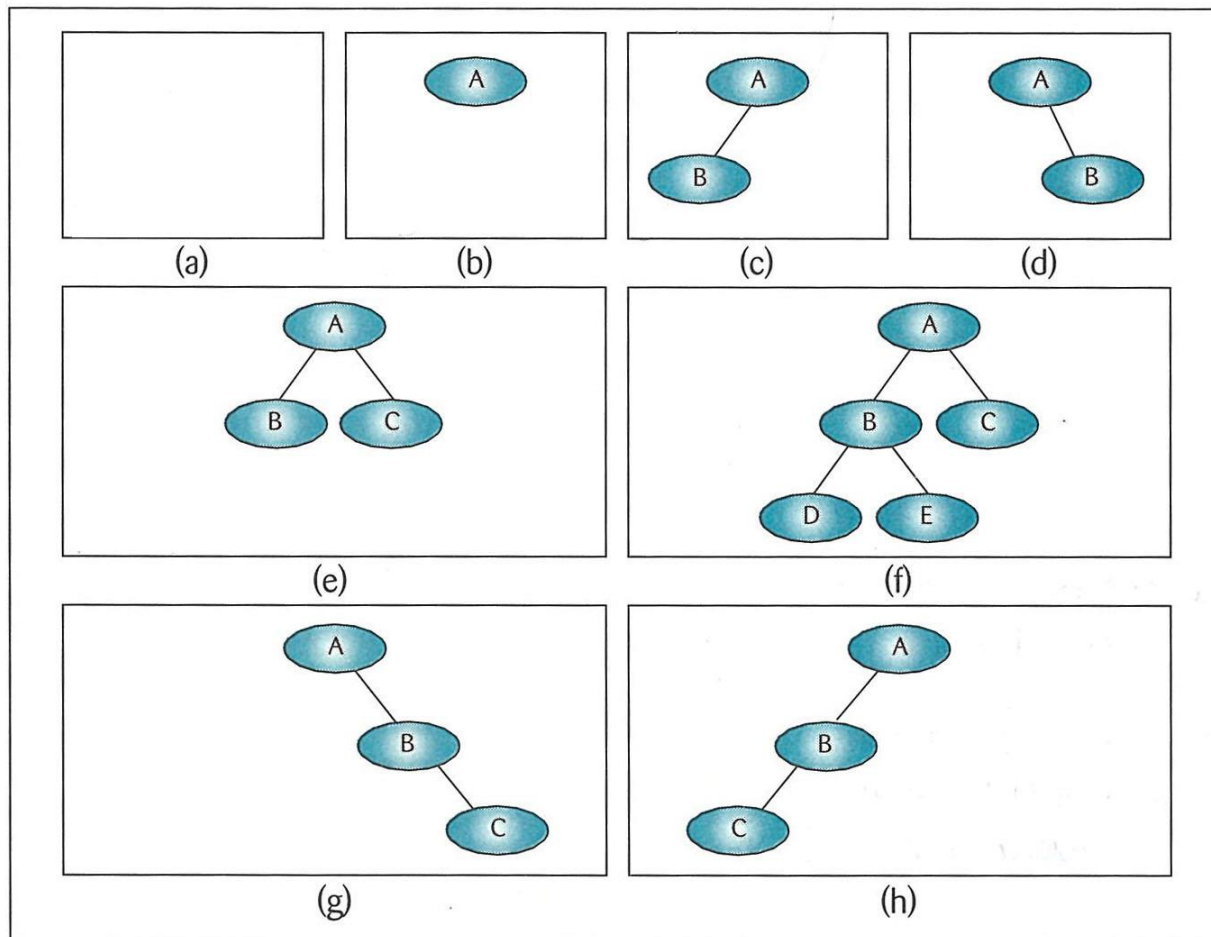
- ▶ 二元樹定義：
- ▶ 1.所有的節點(node)至多只能有兩個子樹，稱二元樹。
- ▶ 2.左邊稱為左子樹，右邊稱為右子樹。
- ▶ 3.樹不一定要對稱。



Binary tree

二元搜尋樹(Binary Search Tree)

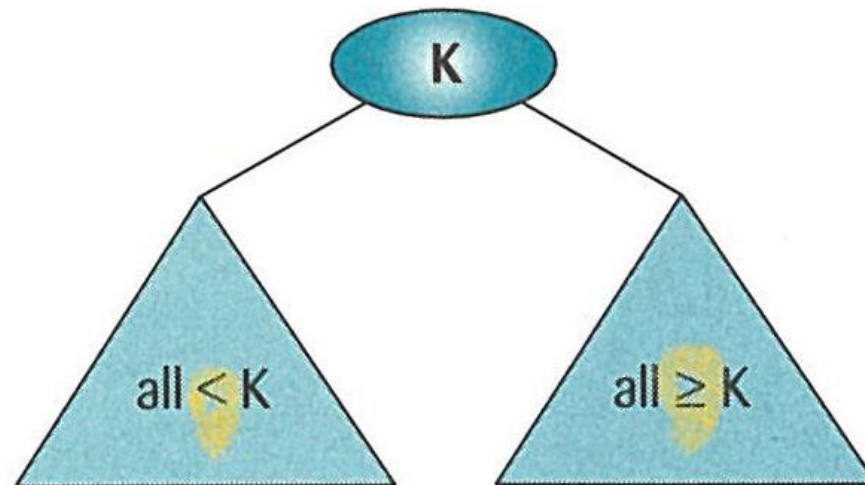
► 一些二元樹的例子：



A collection of binary trees

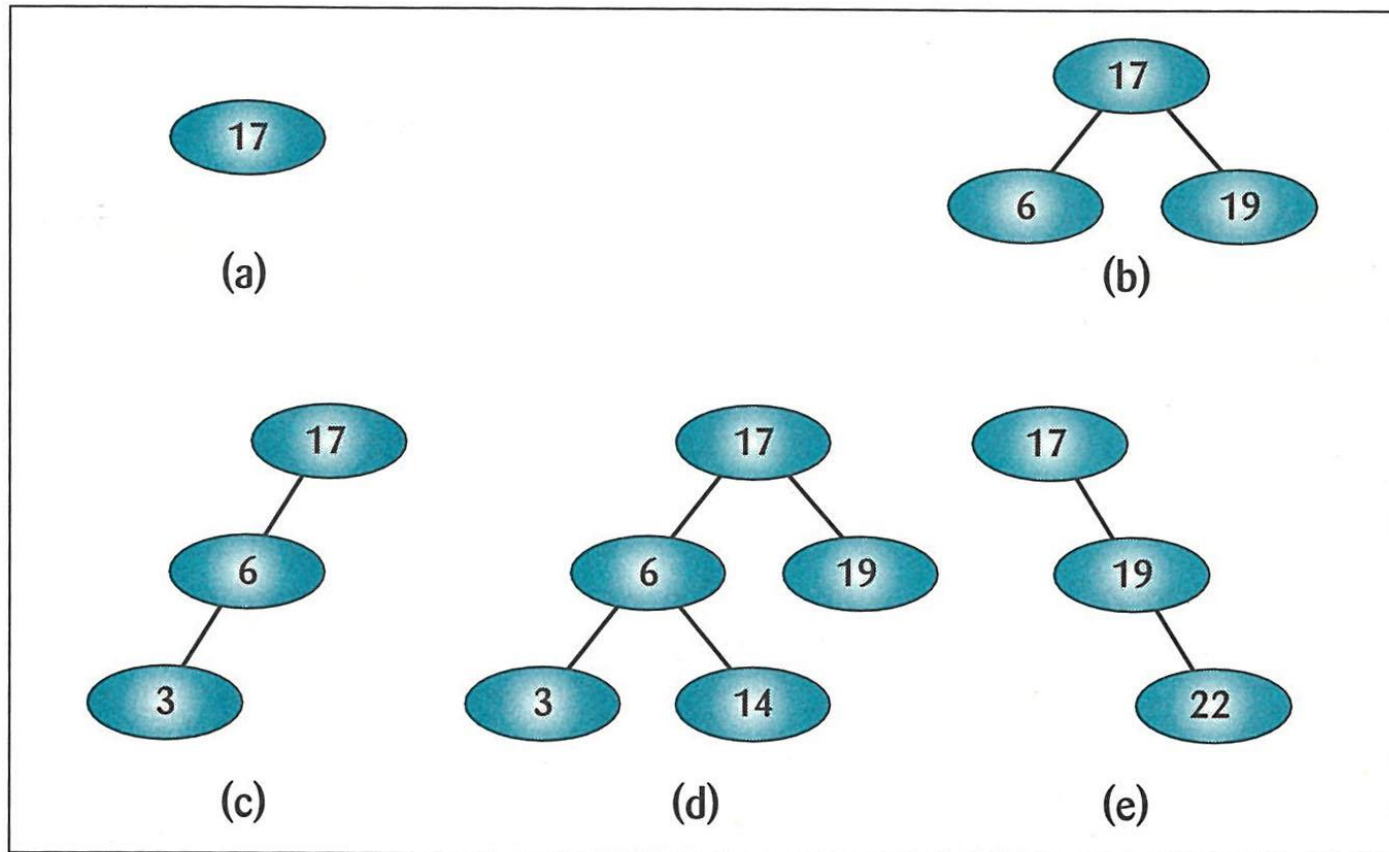
二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹(Binary Search Tree, BST)是一個二元樹，並具有下列性質：
- ▶ 1.任一左子樹都比它的根來的小。
- ▶ 2.任一右子樹都大於或等於它的根。
- ▶ 3.每一個子樹亦是一個二元搜尋樹。



Ch11.資料結構 – 二元搜尋樹(Binary Search Tree)

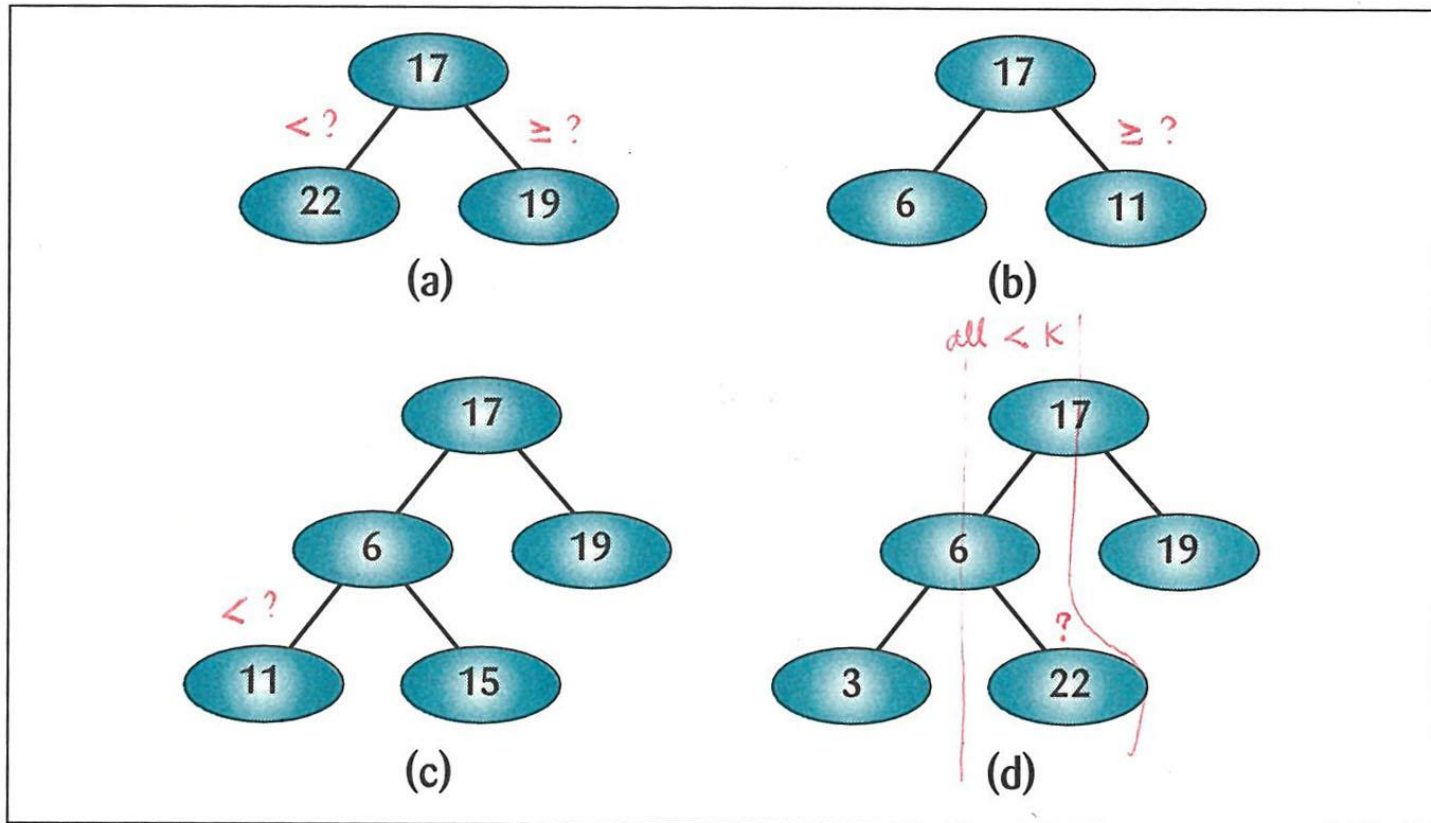
► 一些二元搜尋樹的例子：



Binary search trees

二元搜尋樹(Binary Search Tree)

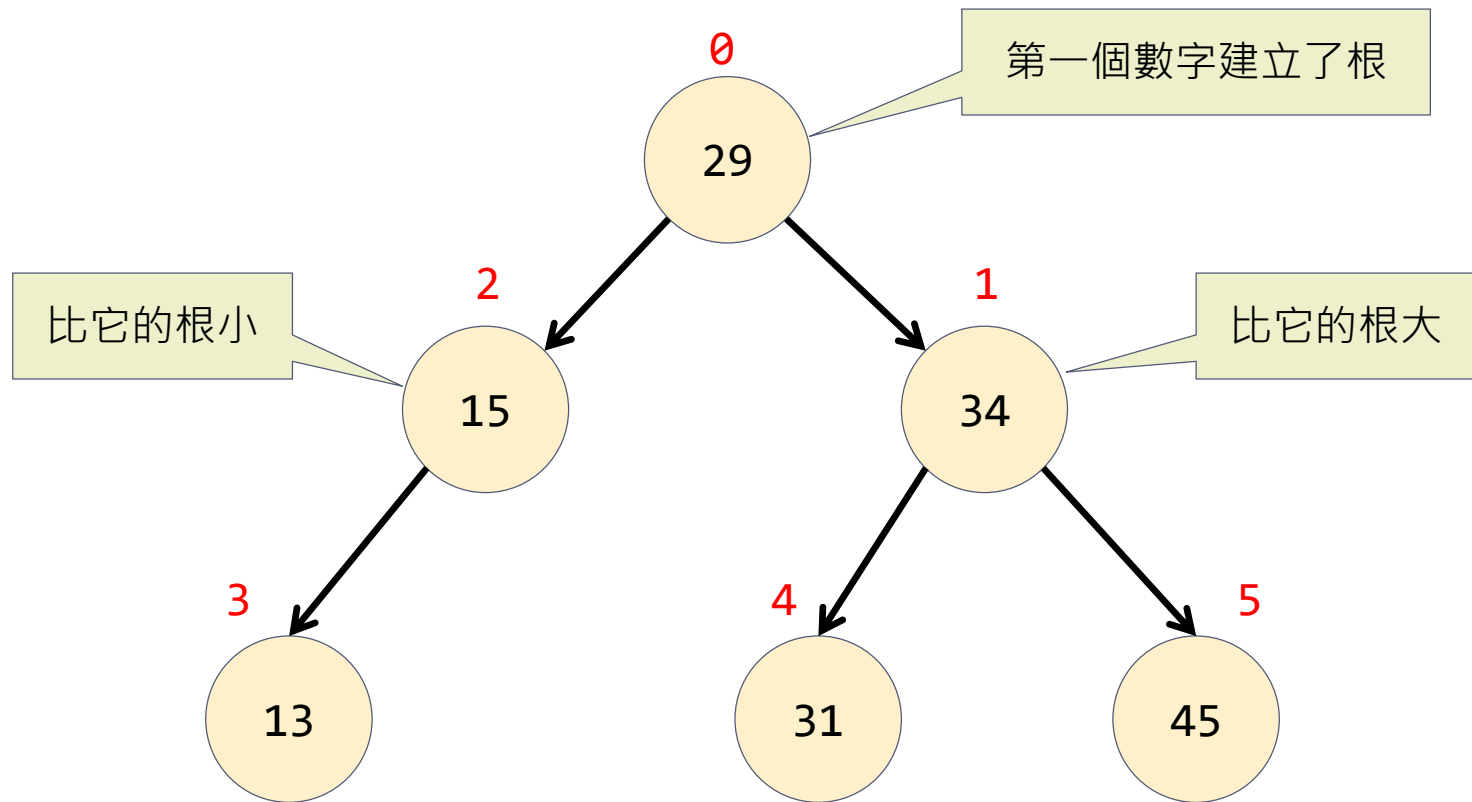
- 一些無效的二元搜尋樹的例子：



Invalid binary search trees

二元搜尋樹(Binary Search Tree)

- ▶ 如何建立一個二元搜尋樹
- ▶ 假設輸入數字順序如下：29、34、15、13、31、45



二元搜尋樹(Binary Search Tree)

- 在這裡我們使用結構來表示一個節點：

Structure node

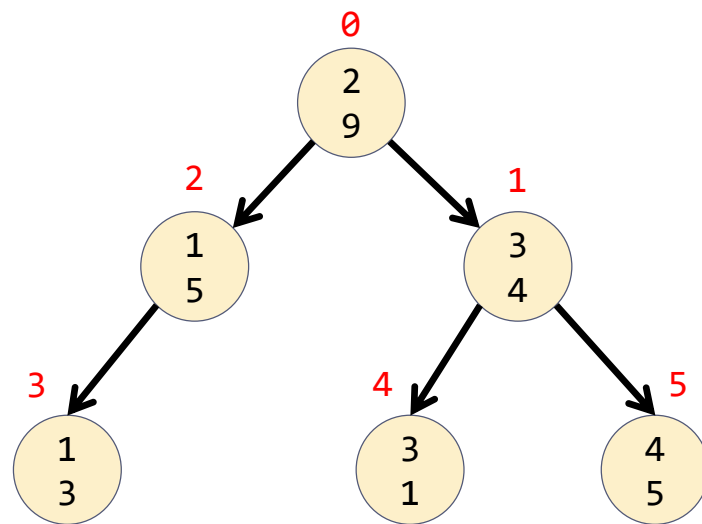
```
Dim value As Integer '節點的值
Dim num As Integer   '節點的編號
Dim L As Integer     '節點的左子節點編號
Dim R As Integer     '節點的右子節點編號
```

End Structure

- 所以一棵樹可用一個結構陣列來表示：

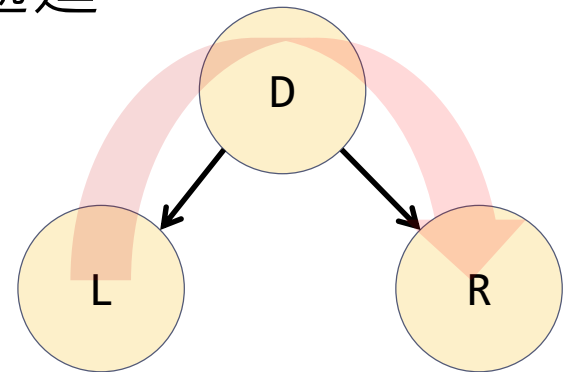
Tree()	value	num	L	R
T(0)	29	0	2	1
T(1)	34	1	4	5
T(2)	15	2	3	-1
T(3)	13	3	-1	-1
T(4)	31	4	-1	-1
T(5)	45	5	-1	-1

- 1表示無子節點。



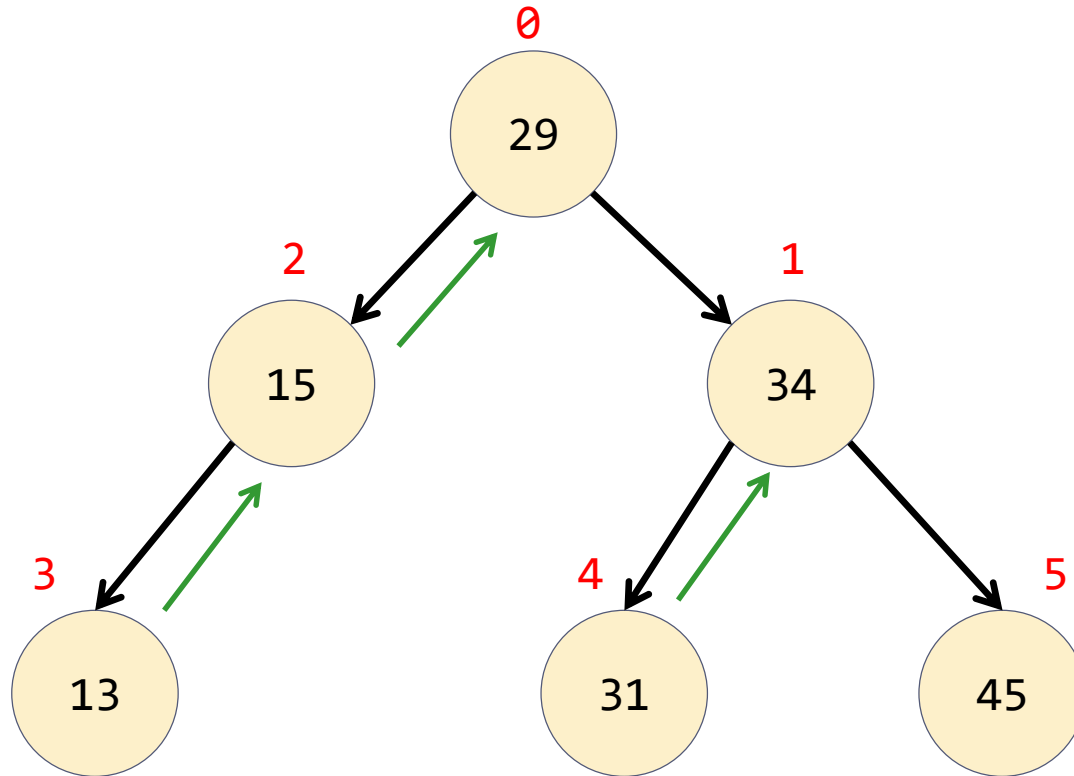
二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹走訪(列出所有資料)，有三種方式：
 - ▶ DLR: 前序走訪 (Preorder Traversal)
 - ▶ **LDR: 中序走訪(Inorder Traversal)**
 - ▶ LRD: 後序走訪(Postorder Traversal)
 - ▶ (D: root, L: 左子樹, R: 右子樹)
- ▶ 以LDR為例，是先走訪左子樹(直到沒有左子樹為止)，根結點，然後右子樹。如果右子樹底下還有左子樹，擇期左子樹要優先訪，這是一個遞迴。



二元搜尋樹(Binary Search Tree)

► LDR中序走訪 (Inorder Traversal) :



輸出結果： 13 15 29 31 34 45

二元搜尋樹(Binary Search Tree)

▶ 範例程式，結構宣告：

```
Structure node
```

```
    Dim value As Integer '節點的值  
    Dim num As Integer '節點的編號  
    Dim L As Integer '節點的左子節點編號  
    Dim R As Integer '節點的右子節點編號
```

```
End Structure
```

```
Dim T(0) As node ' "節點"的陣列，一棵"樹"
```

二元搜尋樹(Binary Search Tree)

▶ 主程式：

```
Sub Main() '主程式
    Dim x, y As Integer
    T(0).num = -1 '先指定成-1，表示尚未建立root
    While True
        Console.WriteLine("[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:")
        x = Val(Console.ReadLine)
        Select Case x
            Case 1 '加入一個節點到樹
                Console.WriteLine("請輸入要加入的數字:")
                y = Console.ReadLine
                If T(0).num = -1 Then '空樹，此為第一個節點root，將此node加入binary tree T中
                    T(0).value = y
                    T(0).num = 0
                    T(0).L = -1 '用-1代表左無子樹
                    T(0).R = -1 '用-1代表右無子樹
                Else
                    insert(0, y) '試著找到適合的地方，將輸入的數字插入
                End If
            Case 2 '從樹中搜尋一個資料
                Console.WriteLine("請輸入要搜尋的數字:")
                y = Console.ReadLine
                Console.WriteLine("從NODE(0)開始尋找...")
                searchValueFromTree(0, y)
            Case 3 '中序查看Tree
                inorderShowTree(0)
            Case 4 '離開程式
                Exit While
        End Select
    End While
End Sub
```

二元搜尋樹(Binary Search Tree)

- 二元搜尋樹的輸入：將一個數字input_value輸入到某一節點index。

```
Sub insert(index, input_value) '加入一個節點到樹
    Dim U As Integer
    '若輸入的數字比此node所儲存的值還小，則針對left chile tree
    If input_value < T(index).value Then
        '若left chile tree存在，試著在left sub-tree中找到適合的位置插入輸入的數字
        If T(index).L > -1 Then
            insert(T(index).L, input_value)
        Else '若left chile tree不存在，將此node產生一個新的left chile node並儲存此輸入的數字
            U = UBound(T) + 1
            ReDim Preserve T(U) '產生一個新的node
            T(U).value = input_value : T(U).num = U
            T(U).L = -1 : T(U).R = -1
            T(index).L = U
            Console.WriteLine("加入新node:NODE(" & U & ")(此node為NODE(" & T(index).num & ")的left chile node).")
        End If
    Else '若輸入的數字比此node所儲存的值還大，則針對right chile tree
        If T(index).R > -1 Then
            insert(T(index).R, input_value)
        Else '若right chile tree不存在，將此node產生一個新的right chile node並儲存此輸入的數字
            U = UBound(T) + 1
            ReDim Preserve T(U) '產生一個新的node
            T(U).value = input_value : T(U).num = U
            T(U).L = -1 : T(U).R = -1
            T(index).R = U
            Console.WriteLine("加入新node:NODE(" & U & ")(此node為NODE(" & T(index).num & ")的right chile node).")
        End If
    End If
End Sub
```

二元搜尋樹(Binary Search Tree)

▶ 二元搜尋樹的搜尋：

```
Sub searchValueFromTree(index, input_value) '從樹中搜尋一個資料
    Console.WriteLine("NODE(" & T(index).num & ")所儲存的數字為" & T(index).value)
    If input_value < T(index).value Then '若比此node所儲存的數字還小，則找left sub-tree
        Console.WriteLine("尋找NODE(" & T(index).num & ")的left child node:NODE(" &
            T(index).L & ")")
        If T(index).L = -1 Then
            Console.WriteLine("NODE(" & T(index).num & ")的left child node不存在，找不到" &
                input_value & "這個數字!")
        Else
            searchValueFromTree(T(index).L, input_value)
        End If
    ElseIf input_value > T(index).value Then '若比此node所儲存的數字還大，則找right sub-tree
        Console.WriteLine("尋找NODE(" & T(index).num & ")的right child node:NODE(" &
            T(index).R & ")")
        If T(index).R = -1 Then
            Console.WriteLine("NODE(" & T(index).num & ")的right child node不存在，找不到" &
                input_value & "這個數字!")
        Else
            searchValueFromTree(T(index).R, input_value)
        End If
    Else '此node所儲存的數字與input_value相同
        Console.WriteLine(input_value & "這個數字存在tree中(位置在NODE(" & T(index).num & "))!")
    End If
End Sub
```



二元搜尋樹(Binary Search Tree)

- ▶ 二元搜尋樹的讀取資料：

```
Sub inorderShowTree(index) '中序查看Tree
    If T(index).L > -1 Then '追左樹
        inorderShowTree(T(index).L)
    End If
    Console.WriteLine("NODE(" & T(index).num & ")所儲存的數字為" &
        T(index).value & ",left chile node為NODE(" &
        T(index).L & "),right chile node為NODE(" &
        T(index).R & ")") '中節點
    If T(index).R > -1 Then '追右樹
        inorderShowTree(T(index).R)
    End If
End Sub
```

- ▶ 要讀出整棵樹裡面各節點的數字，我們用採中序式走訪。

二元搜尋樹(Binary)

▶ 執行結果：

```
file:///C:/Users/Orion Liu/documents/visual studio 2010/Projects/ConsoleApplication...
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:16
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:88
加入新node:NODE(1)(此node為NODE(0)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:49
加入新node:NODE(2)(此node為NODE(1)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:8
加入新node:NODE(3)(此node為NODE(0)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:65
加入新node:NODE(4)(此node為NODE(2)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:99
加入新node:NODE(5)(此node為NODE(1)的right chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:1
請輸入要加入的數字:4
加入新node:NODE(6)(此node為NODE(3)的left chile node).
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:2
請輸入要搜尋的數字:10
從NODE(0)開始尋找...
NODE(0)所儲存的數字為16
尋找NODE(0)的left child node:NODE(3)
NODE(3)所儲存的數字為8
尋找NODE(3)的right child node:NODE(-1)
NODE(3)的right child node不存在，找不到10這個數字!
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:2
請輸入要搜尋的數字:8
從NODE(0)開始尋找...
NODE(0)所儲存的數字為16
尋找NODE(0)的left child node:NODE(3)
NODE(3)所儲存的數字為8
8這個數字存在tree中( 位置在NODE(3))!
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:3
NODE(6)所儲存的數字為4,left chile node為NODE(-1),right chile node為NODE(-1)
NODE(3)所儲存的數字為8,left chile node為NODE(6),right chile node為NODE(-1)
NODE(0)所儲存的數字為16,left chile node為NODE(3),right chile node為NODE(1)
NODE(2)所儲存的數字為49,left chile node為NODE(-1),right chile node為NODE(4)
NODE(4)所儲存的數字為65,left chile node為NODE(-1),right chile node為NODE(-1)
NODE(1)所儲存的數字為88,left chile node為NODE(2),right chile node為NODE(5)
NODE(5)所儲存的數字為99,left chile node為NODE(-1),right chile node為NODE(-1)
[Tree] 1.加入數字 2.搜尋數字 3.中序查看Tree 4.離開程式 || 請選擇功能:4
```

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

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

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

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

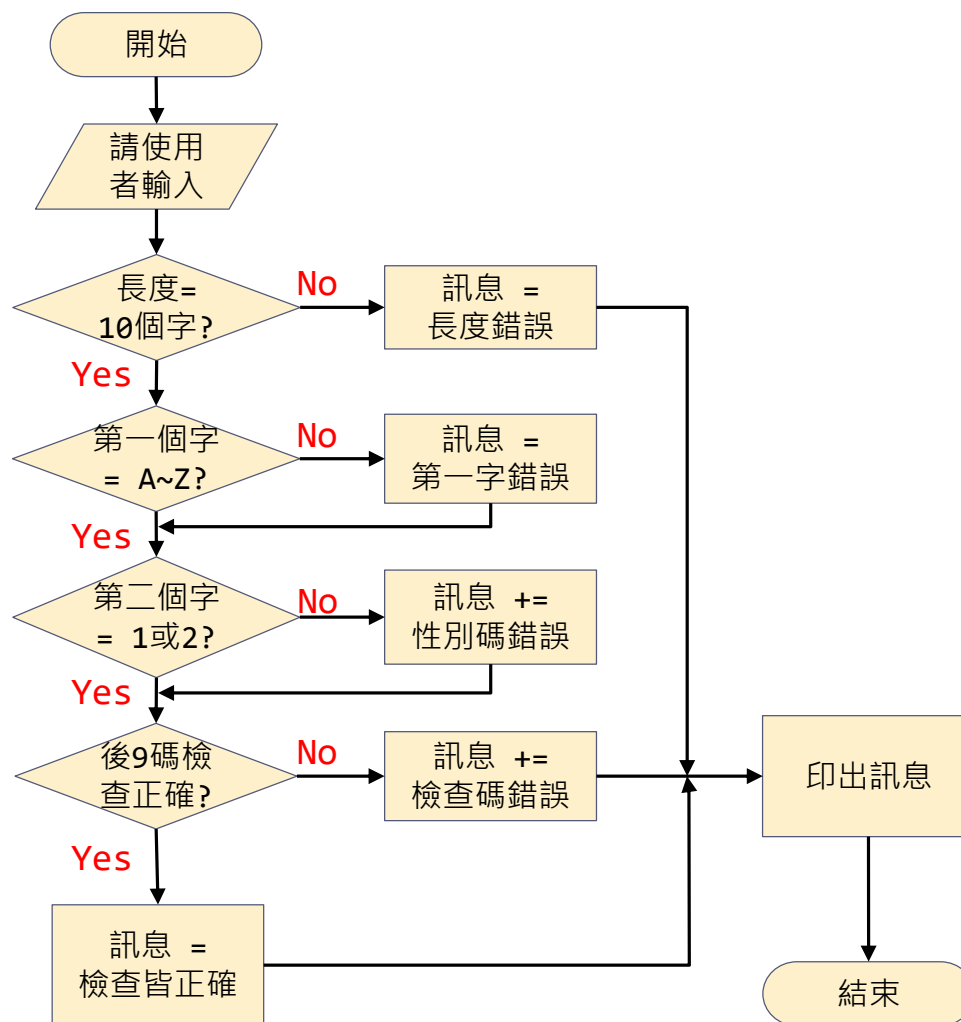
- ▶ 由模數減去餘數得檢查碼

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

- ▶ 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"

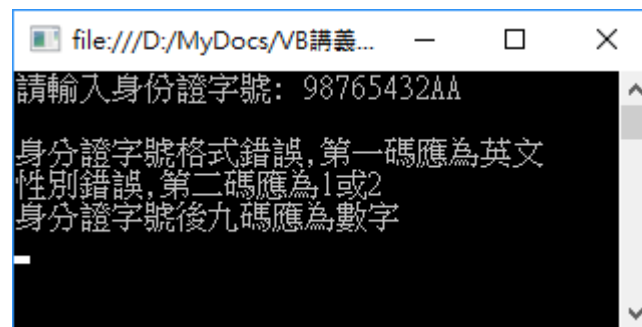
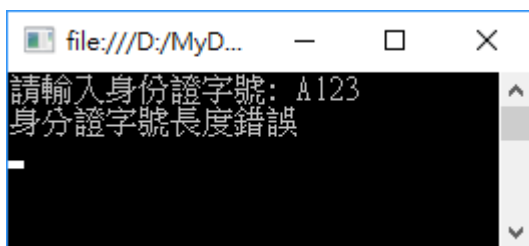
    Console.Write("請輸入身分證字號: ")
    ID = Console.ReadLine().ToUpper
    If Len(ID) <> 10 Then '檢查長度是否為10個字
        ErrMsg = "身分證字號長度錯誤"
    Else '檢查第一碼是否為英文字
        If Mid(ID, 1, 1) < "A" Or Mid(ID, 1, 1) > "Z" Then
            ErrMsg = ErrMsg & vbCrLf & "身分證字號格式錯誤,第一碼應為英文"
        End If
        If Mid(ID, 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
```

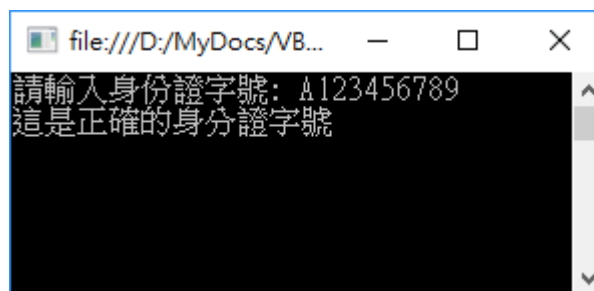
綜合練習一、身分證字號檢查

▶ 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()
```



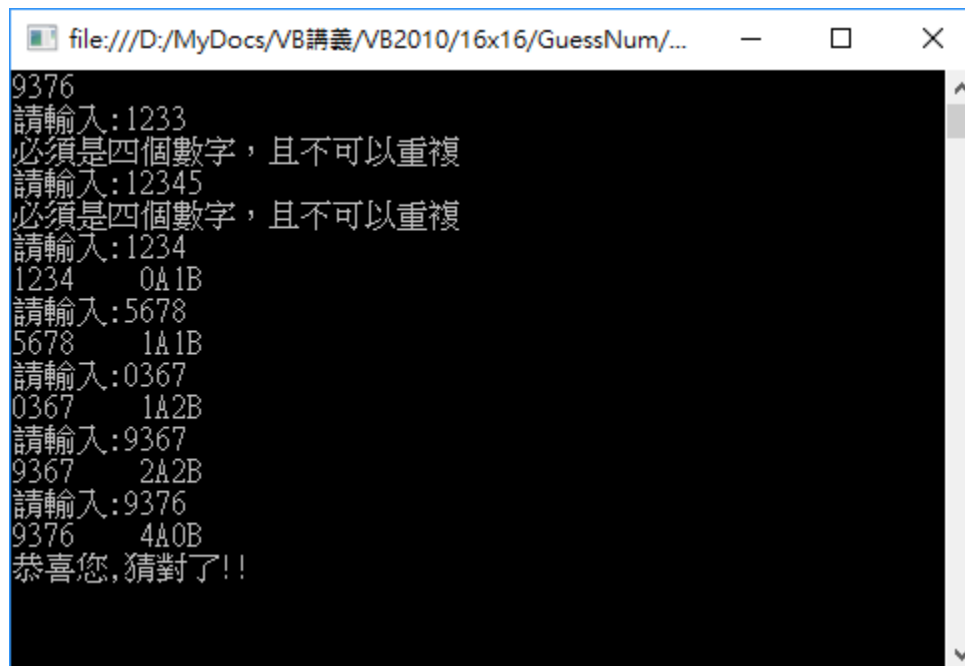
綜合練習二、猜數字遊戲

- ▶ 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 '存放答案

    0 個參考
    Sub Main()
        Randomize()
        Dim X As String
        '產生題目，確保不重複的數字
        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
        '請使用者輸入直到答對
        Do
            Console.WriteLine("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.WriteLine(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 = q2 Or a3 = q1 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 = q2 Or a4 = q3 Or a4 = q1) 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
```

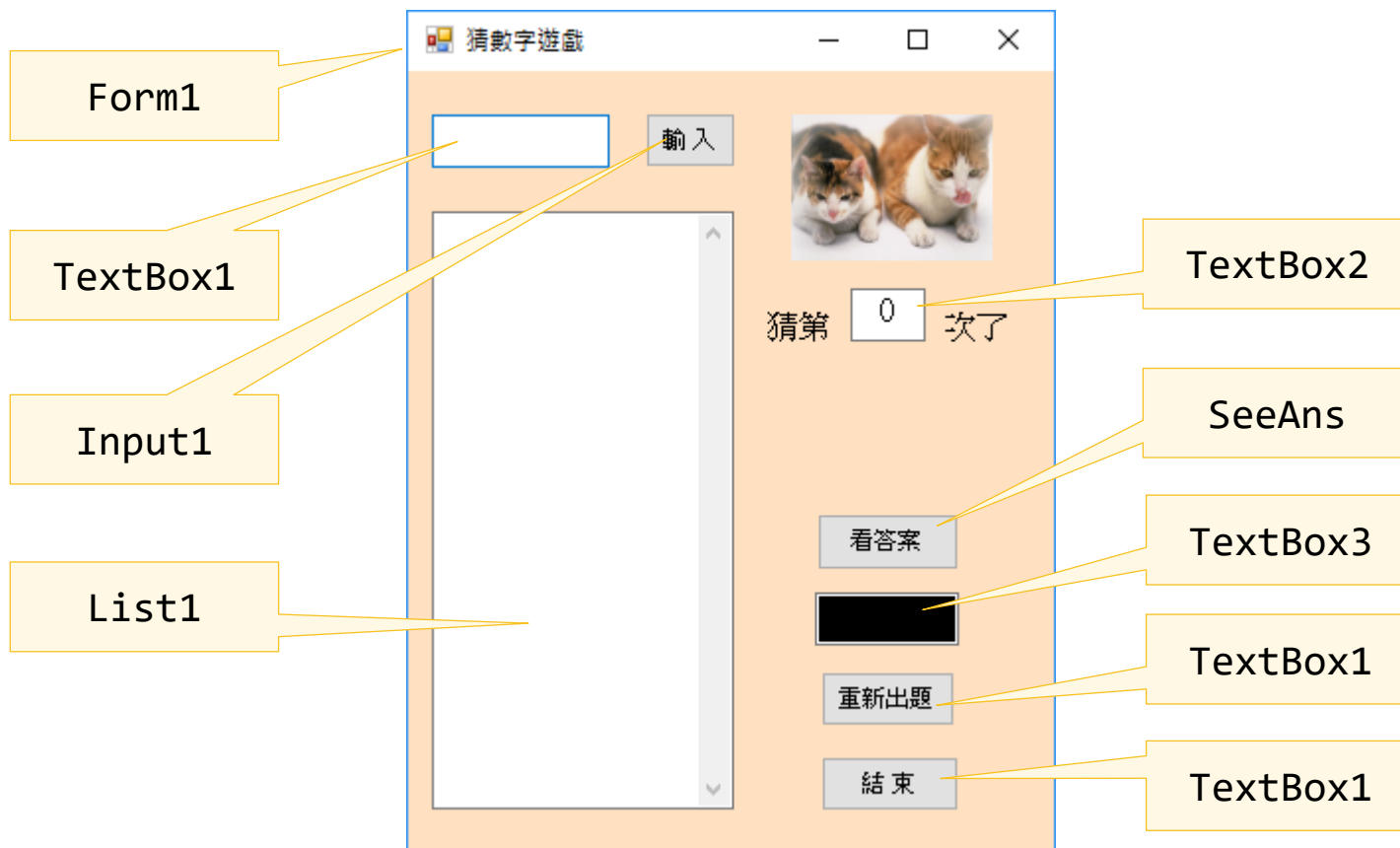
綜合練習二、猜數字遊戲

▶ 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：



綜合練習三、猜數字遊戲視窗版

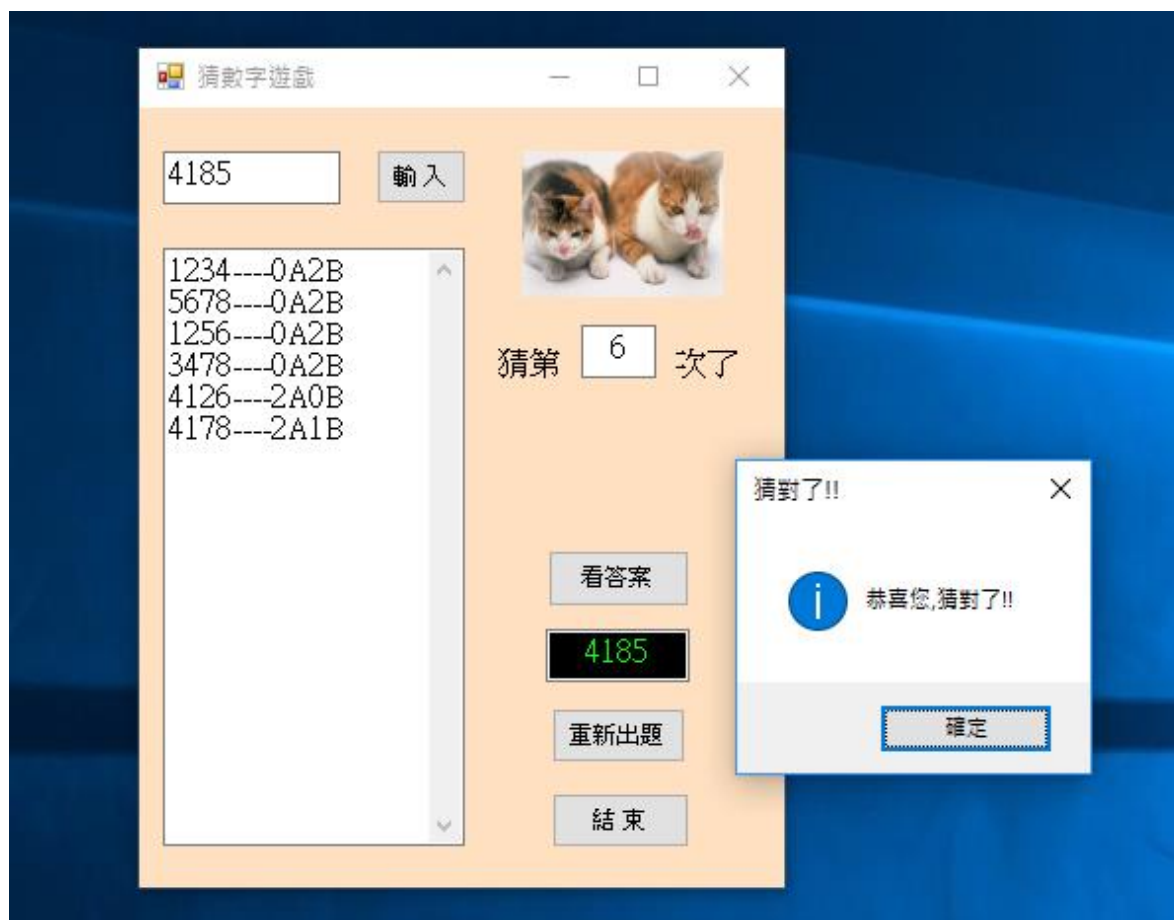
▶ 參考程式：(請參閱講義)

```

1  > 1000000000
2  Public Class Form1
3      Dim A As Integer '傳入變數A 傳入時數
4      Dim B1, B2, B3, B4 As Integer '傳入變數B 傳入時數
5      Dim B5, B6, B7, B8, B9 As Integer '傳入變數B 傳入時數
6
7      '0 個參數
8      Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
9          Call getrand() '隨手生成一個隨機數
10     End Sub
11
12     '0 個參數
13     Private Sub Inputs_Click(sender As Object, e As EventArgs) Handles Inputs.Click
14         Call check() '輸入 - 按鍵 傳入隨機輸入測試驗證
15     End Sub
16
17     '0 個參數
18     Private Sub TextBox1_TextChanged(sender As Object, e As EventArgs) Handles TextBox1.TextChanged
19         '輸入 - 文字框內容輸入
20         If TextBox1.Text.Length > 0 Then '傳入隨機輸入測試驗證
21             Call check()
22         End If
23     End Sub
24
25     '0 個參數
26     Private Sub Sema_Click(sender As Object, e As EventArgs) Handles Sema.Click
27         TextBox.Text = q1 & q2 & q3 & q4 '輸出結果
28     End Sub
29
30     '0 個參數
31     Private Sub Buttons_Click(sender As Object, e As EventArgs) Handles Buttons.Click
32         TextBox.Enabled = True '輸入 - 文字框內容輸入
33         Sema.Enabled = True '輸入 - 按鍵
34         Inputs.Enabled = True '輸入 - 按鍵
35         TextBox.Focus() '輸入 - 文字框內容輸入
36         Call clear() '清除輸入內容
37         Call getrand() '隨手生成一個隨機數
38     End Sub
39
40     '0 個參數
41     Private Sub Buttons_Click(sender As Object, e As EventArgs) Handles Buttons.Click
42         End Sub
43
44 '2 個參數
45 Private Sub check() '傳入輸入測試驗證
46     Dim A As Integer
47     A = A + 1
48     If A < 10 Then '傳入輸入測試驗證
49         Call getrand()
50     End If
51     If A < 10 Then '傳入輸入測試驗證
52         Call getrand()
53     End If
54     If A < 10 Then '傳入輸入測試驗證
55         Call getrand()
56     End If
57     If A < 10 Then '傳入輸入測試驗證
58         Call getrand()
59     End If
60     If A < 10 Then '傳入輸入測試驗證
61         Call getrand()
62     End If
63     If A < 10 Then '傳入輸入測試驗證
64         Call getrand()
65     End If
66     If A < 10 Then '傳入輸入測試驗證
67         Call getrand()
68     End If
69     If A < 10 Then '傳入輸入測試驗證
70         Call getrand()
71     End If
72     If A < 10 Then '傳入輸入測試驗證
73         Call getrand()
74     End If
75     If A < 10 Then '傳入輸入測試驗證
76         Call getrand()
77     End If
78     If A < 10 Then '傳入輸入測試驗證
79         Call getrand()
80     End If
81     If A < 10 Then '傳入輸入測試驗證
82         Call getrand()
83     End If
84     If A < 10 Then '傳入輸入測試驗證
85         Call getrand()
86     End If
87     If A < 10 Then '傳入輸入測試驗證
88         Call getrand()
89     End If
90     If A < 10 Then '傳入輸入測試驗證
91         Call getrand()
92     End If
93     If A < 10 Then '傳入輸入測試驗證
94         Call getrand()
95     End If
96     If A < 10 Then '傳入輸入測試驗證
97         Call getrand()
98     End If
99     If A < 10 Then '傳入輸入測試驗證
100        Call getrand()
101    End If
102    If A < 10 Then '傳入輸入測試驗證
103        Call getrand()
104    End If
105    If A < 10 Then '傳入輸入測試驗證
106        Call getrand()
107    End If
108    If A < 10 Then '傳入輸入測試驗證
109        Call getrand()
110    End If
111    If A < 10 Then '傳入輸入測試驗證
112        Call getrand()
113    End If
114    If A < 10 Then '傳入輸入測試驗證
115        Call getrand()
116    End If
117    If A < 10 Then '傳入輸入測試驗證
118        Call getrand()
119    End If
120    If A < 10 Then '傳入輸入測試驗證
121        Call getrand()
122    End If
123    If A < 10 Then '傳入輸入測試驗證
124        Call getrand()
125    End If
126    If A < 10 Then '傳入輸入測試驗證
127        Call getrand()
128    End If
129    If A < 10 Then '傳入輸入測試驗證
130        Call getrand()
131    End If
132    If A < 10 Then '傳入輸入測試驗證
133        Call getrand()
134    End If
135    If A < 10 Then '傳入輸入測試驗證
136        Call getrand()
137    End If
138    If A < 10 Then '傳入輸入測試驗證
139        Call getrand()
140    End If
141    If A < 10 Then '傳入輸入測試驗證
142        Call getrand()
143    End If
144    If A < 10 Then '傳入輸入測試驗證
145        Call getrand()
146    End If
147    If A < 10 Then '傳入輸入測試驗證
148        Call getrand()
149    End If
150    If A < 10 Then '傳入輸入測試驗證
151        Call getrand()
152    End If
153    If A < 10 Then '傳入輸入測試驗證
154        Call getrand()
155    End If
156    If A < 10 Then '傳入輸入測試驗證
157        Call getrand()
158    End If
159    If A < 10 Then '傳入輸入測試驗證
160        Call getrand()
161    End If
162    If A < 10 Then '傳入輸入測試驗證
163        Call getrand()
164    End If
165    If A < 10 Then '傳入輸入測試驗證
166        Call getrand()
167    End If
168    If A < 10 Then '傳入輸入測試驗證
169        Call getrand()
170    End If
171    If A < 10 Then '傳入輸入測試驗證
172        Call getrand()
173    End If
174    If A < 10 Then '傳入輸入測試驗證
175        Call getrand()
176    End If
177    If A < 10 Then '傳入輸入測試驗證
178        Call getrand()
179    End If
180    If A < 10 Then '傳入輸入測試驗證
181        Call getrand()
182    End If
183    If A < 10 Then '傳入輸入測試驗證
184        Call getrand()
185    End If
186    If A < 10 Then '傳入輸入測試驗證
187        Call getrand()
188    End If
189    If A < 10 Then '傳入輸入測試驗證
190        Call getrand()
191    End If
192    If A < 10 Then '傳入輸入測試驗證
193        Call getrand()
194    End If
195    If A < 10 Then '傳入輸入測試驗證
196        Call getrand()
197    End If
198    If A < 10 Then '傳入輸入測試驗證
199        Call getrand()
200    End If
201    If A < 10 Then '傳入輸入測試驗證
202        Call getrand()
203    End If
204    If A < 10 Then '傳入輸入測試驗證
205        Call getrand()
206    End If
207    If A < 10 Then '傳入輸入測試驗證
208        Call getrand()
209    End If
210    If A < 10 Then '傳入輸入測試驗證
211        Call getrand()
212    End If
213    If A < 10 Then '傳入輸入測試驗證
214        Call getrand()
215    End If
216    If A < 10 Then '傳入輸入測試驗證
217        Call getrand()
218    End If
219    If A < 10 Then '傳入輸入測試驗證
220        Call getrand()
221    End If
222    If A < 10 Then '傳入輸入測試驗證
223        Call getrand()
224    End If
225    If A < 10 Then '傳入輸入測試驗證
226        Call getrand()
227    End If
228    If A < 10 Then '傳入輸入測試驗證
229        Call getrand()
230    End If
231    If A < 10 Then '傳入輸入測試驗證
232        Call getrand()
233    End If
234    If A < 10 Then '傳入輸入測試驗證
235        Call getrand()
236    End If
237    If A < 10 Then '傳入輸入測試驗證
238        Call getrand()
239    End If
240    If A < 10 Then '傳入輸入測試驗證
241        Call getrand()
242    End If
243    If A < 10 Then '傳入輸入測試驗證
244        Call getrand()
245    End If
246    If A < 10 Then '傳入輸入測試驗證
247        Call getrand()
248    End If
249    If A < 10 Then '傳入輸入測試驗證
250        Call getrand()
251    End If
252    If A < 10 Then '傳入輸入測試驗證
253        Call getrand()
254    End If
255    If A < 10 Then '傳入輸入測試驗證
256        Call getrand()
257    End If
258    If A < 10 Then '傳入輸入測試驗證
259        Call getrand()
260    End If
261    If A < 10 Then '傳入輸入測試驗證
262        Call getrand()
263    End If
264    If A < 10 Then '傳入輸入測試驗證
265        Call getrand()
266    End If
267    If A < 10 Then '傳入輸入測試驗證
268        Call getrand()
269    End If
270    If A < 10 Then '傳入輸入測試驗證
271        Call getrand()
272    End If
273    If A < 10 Then '傳入輸入測試驗證
274        Call getrand()
275    End If
276    If A < 10 Then '傳入輸入測試驗證
277        Call getrand()
278    End If
279    If A < 10 Then '傳入輸入測試驗證
280        Call getrand()
281    End If
282    If A < 10 Then '傳入輸入測試驗證
283        Call getrand()
284    End If
285    If A < 10 Then '傳入輸入測試驗證
286        Call getrand()
287    End If
288    If A < 10 Then '傳入輸入測試驗證
289        Call getrand()
290    End If
291    If A < 10 Then '傳入輸入測試驗證
292        Call getrand()
293    End If
294    If A < 10 Then '傳入輸入測試驗證
295        Call getrand()
296    End If
297    If A < 10 Then '傳入輸入測試驗證
298        Call getrand()
299    End If
300    If A < 10 Then '傳入輸入測試驗證
301        Call getrand()
302    End If
303    If A < 10 Then '傳入輸入測試驗證
304        Call getrand()
305    End If
306    If A < 10 Then '傳入輸入測試驗證
307        Call getrand()
308    End If
309    If A < 10 Then '傳入輸入測試驗證
310        Call getrand()
311    End If
312    If A < 10 Then '傳入輸入測試驗證
313        Call getrand()
314    End If
315    If A < 10 Then '傳入輸入測試驗證
316        Call getrand()
317    End If
318    If A < 10 Then '傳入輸入測試驗證
319        Call getrand()
320    End If
321    If A < 10 Then '傳入輸入測試驗證
322        Call getrand()
323    End If
324    If A < 10 Then '傳入輸入測試驗證
325        Call getrand()
326    End If
327    If A < 10 Then '傳入輸入測試驗證
328        Call getrand()
329    End If
330    If A < 10 Then '傳入輸入測試驗證
331        Call getrand()
332    End If
333    If A < 10 Then '傳入輸入測試驗證
334        Call getrand()
335    End If
336    If A < 10 Then '傳入輸入測試驗證
337        Call getrand()
338    End If
339    If A < 10 Then '傳入輸入測試驗證
340        Call getrand()
341    End If
342    If A < 10 Then '傳入輸入測試驗證
343        Call getrand()
344    End If
345    If A < 10 Then
```

綜合練習三、猜數字遊戲視窗版

► 執行結果：



休息一下~~

