



# Visual Basic程式語言 part1

劉和師

2019/02/15

# Visual Basic簡介

---

- ▶ **BASIC**（來自英語：**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode的縮寫），是一種直譯式程式設計語言。名稱字面為意思「初學者的全方位符式指令程式碼」，設計給初學者使用的程式語言。
- ▶ 電腦由文字介面(DOS)進入到圖形介面(GUI)後，這個語言沉寂了一段時間。
- ▶ 後來由微軟公司開發，包含環境的事件驅動等功能，讓它可以在視窗環境下執行，稱為**Visual Basic**（**VB**）。



# Visual Basic簡介

---

- ▶ 目前最新的環境是Visual Studio 2017版(2019年)
  - ▶ 下載：<https://visualstudio.microsoft.com/zh-hant/vs/>



- ▶ 不過因為它包含了豐富的語言與工具支援，初學者如果不想IDE環境太複雜，可以使用VB2010 Express。
  - ▶ 版本太舊，已無法下載，需要者請洽老師。

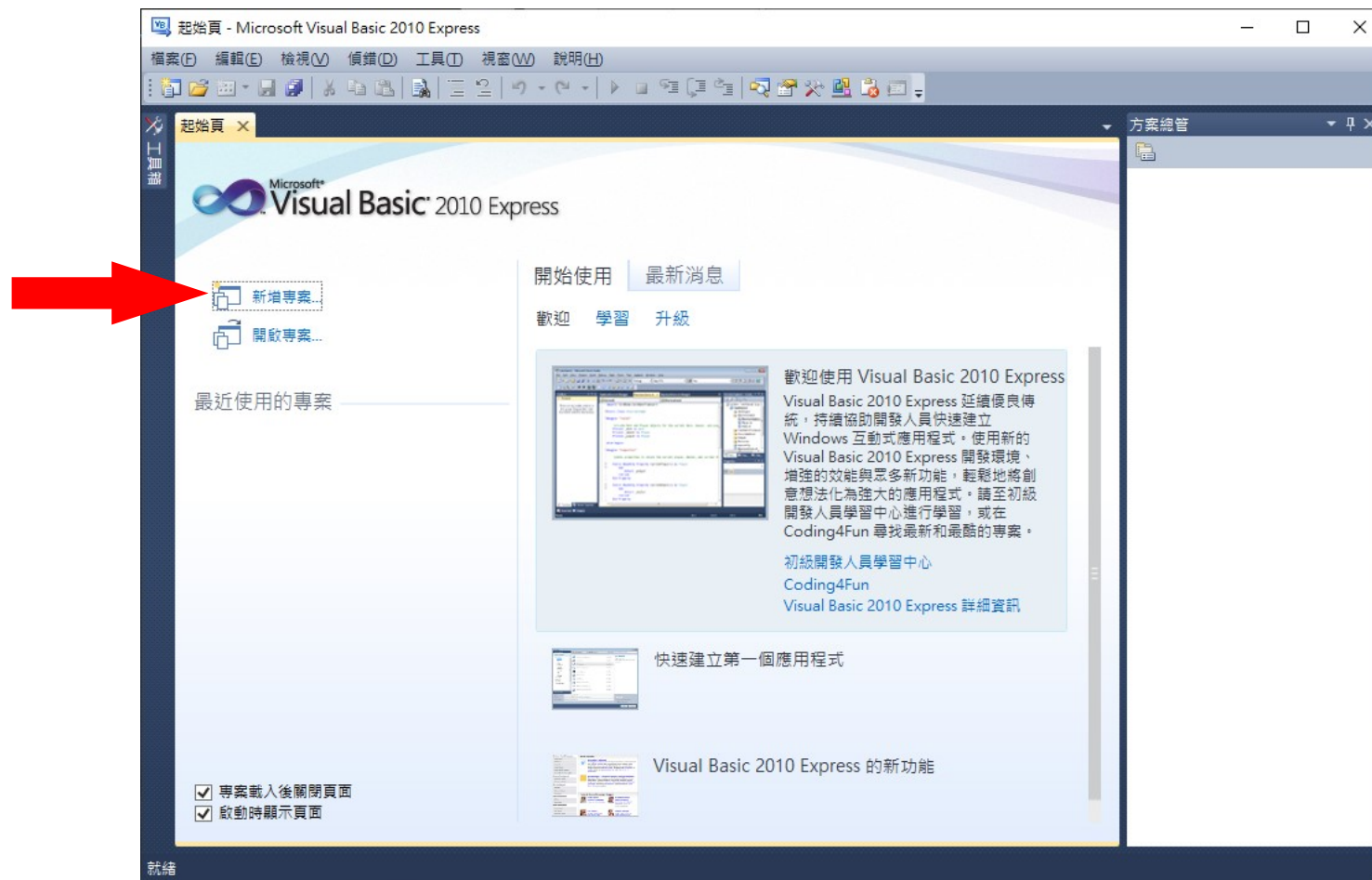


- ▶ 這些工具軟體都是免費的。



# Visual Basic簡介

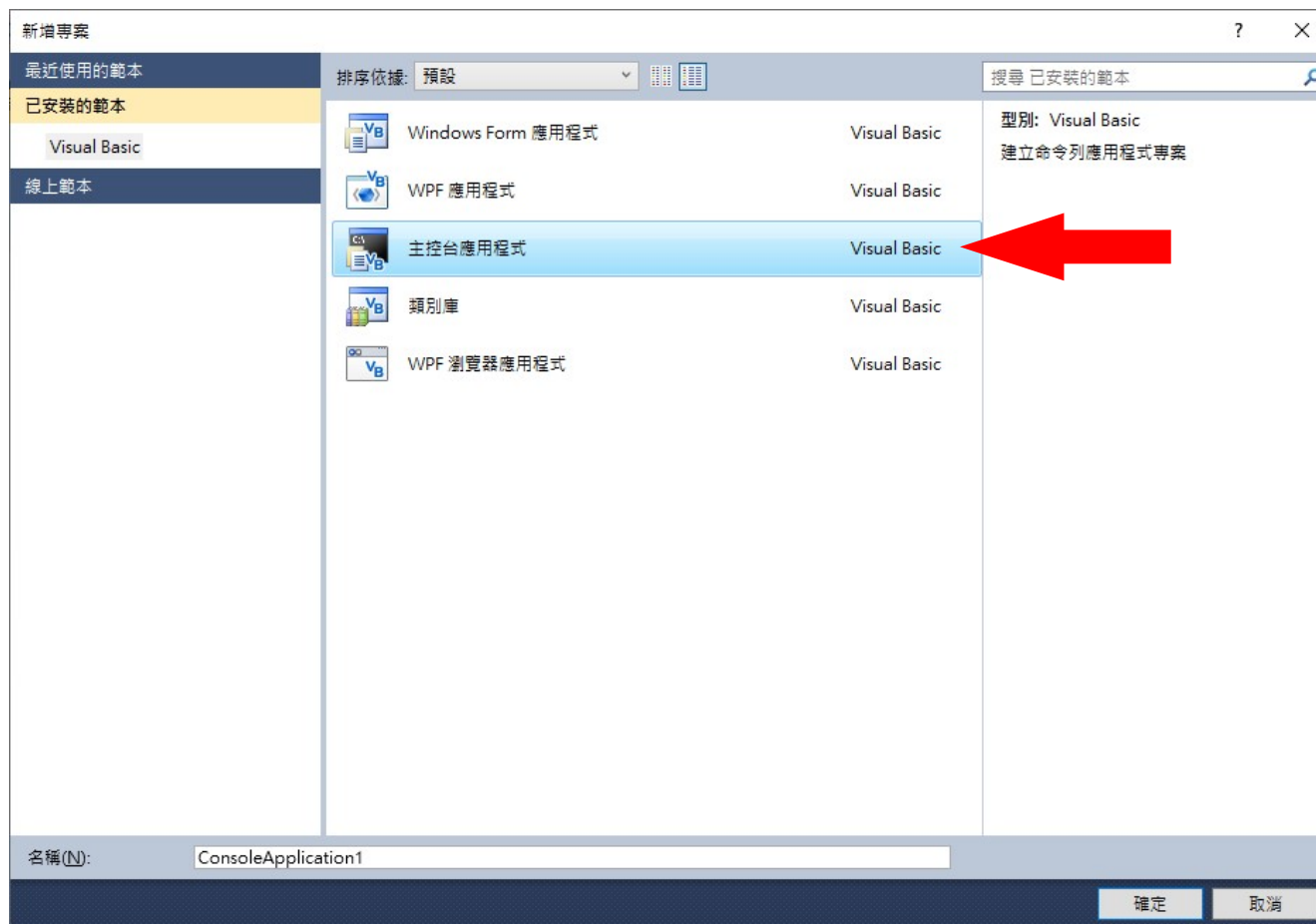
## ▶ VB2010起始畫面





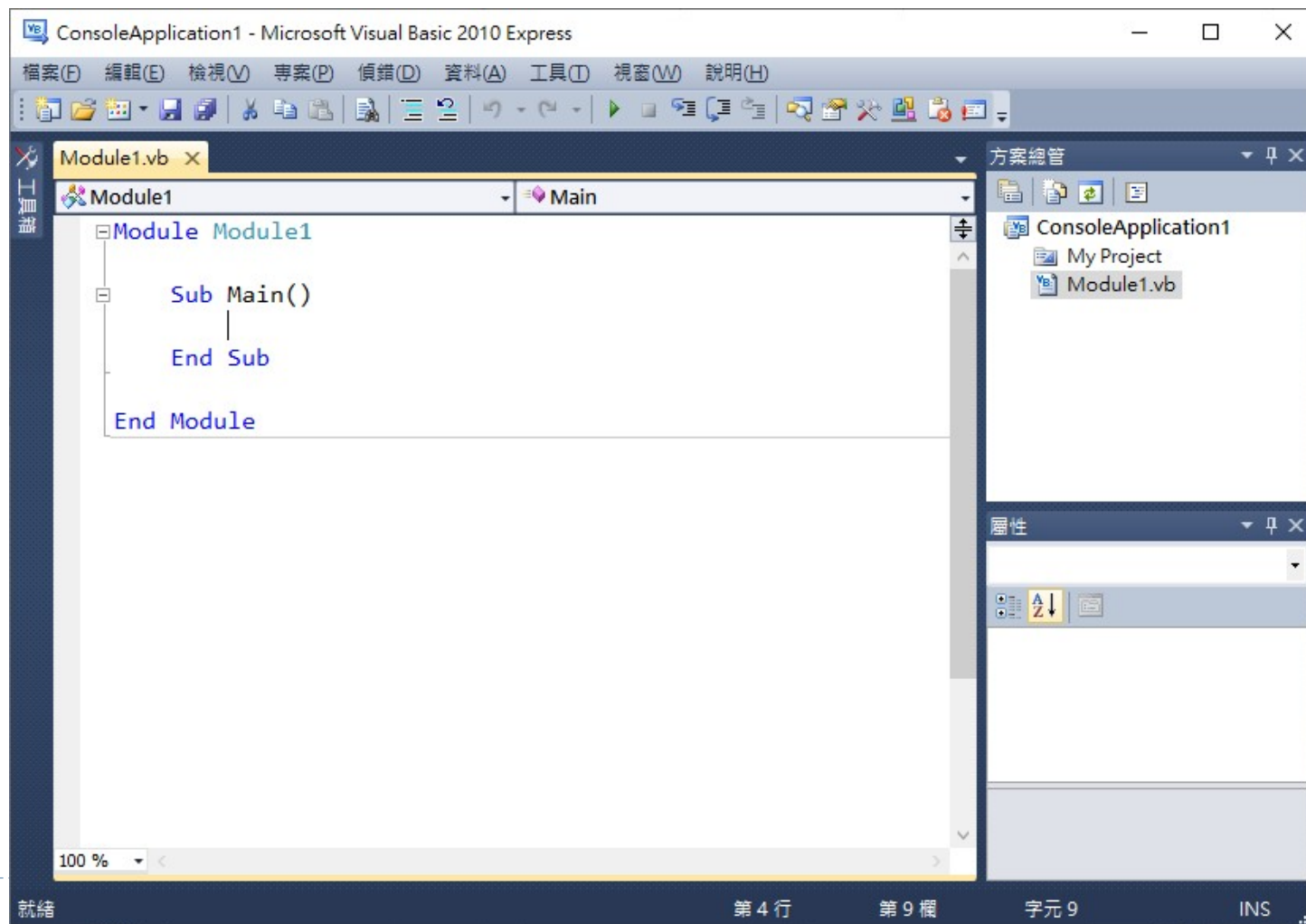
# Visual Basic簡介

- ▶ 我們大部分時候會使用「主控台應用程式」模式



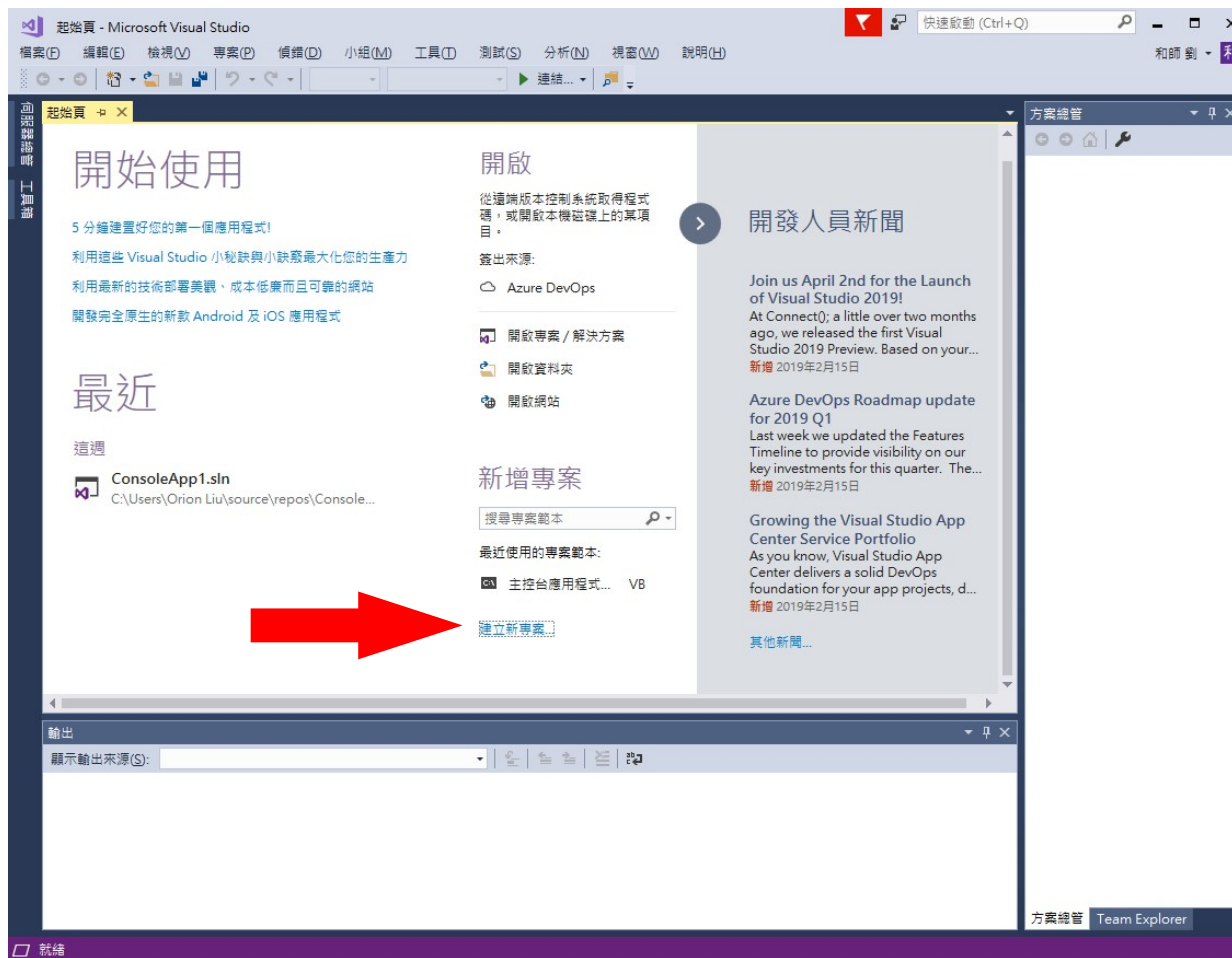
# Visual Basic簡介

## ▶ 進入程式撰寫畫面



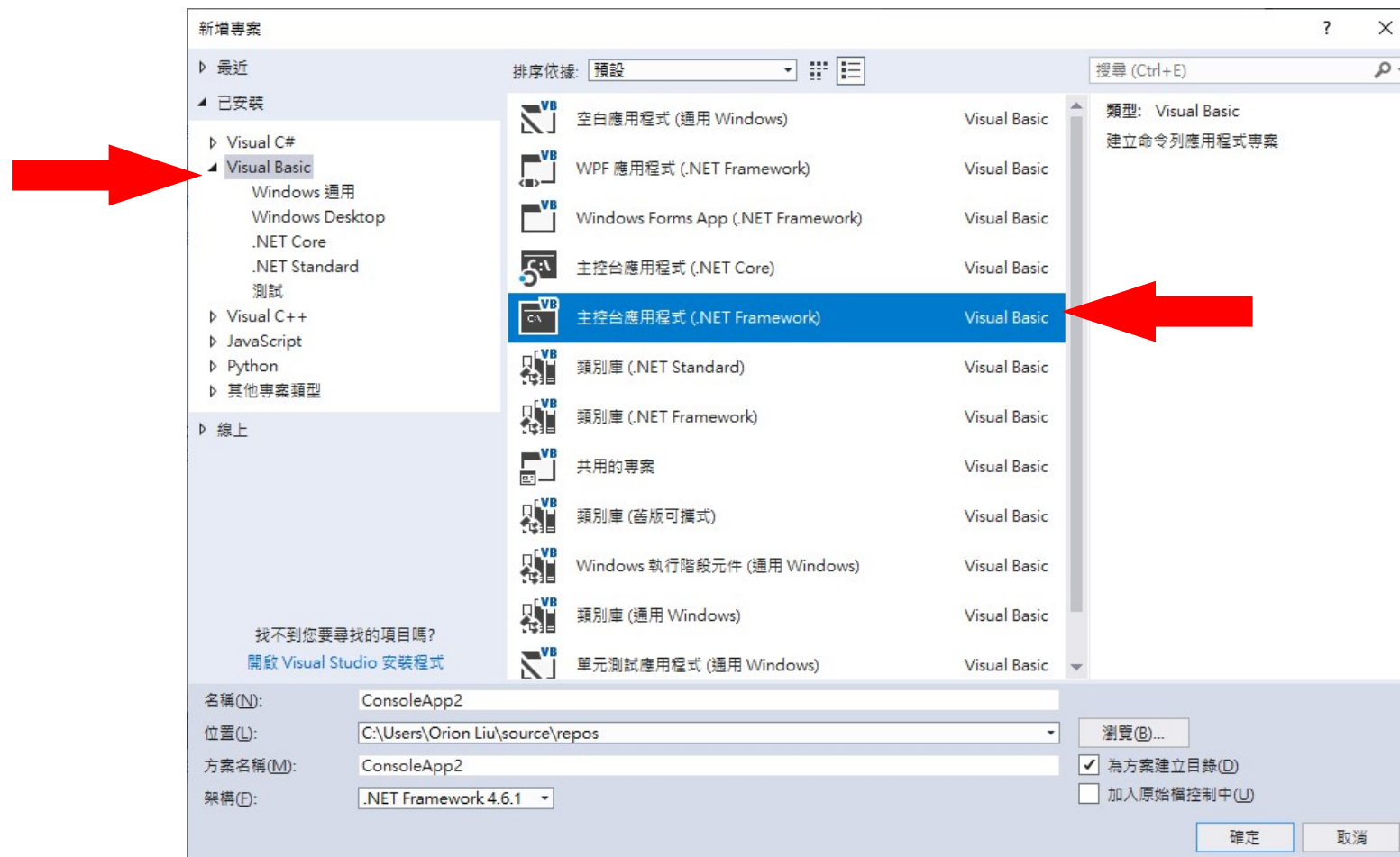
# Visual Basic 簡介

## ► Visual Studio 2017 起始畫面



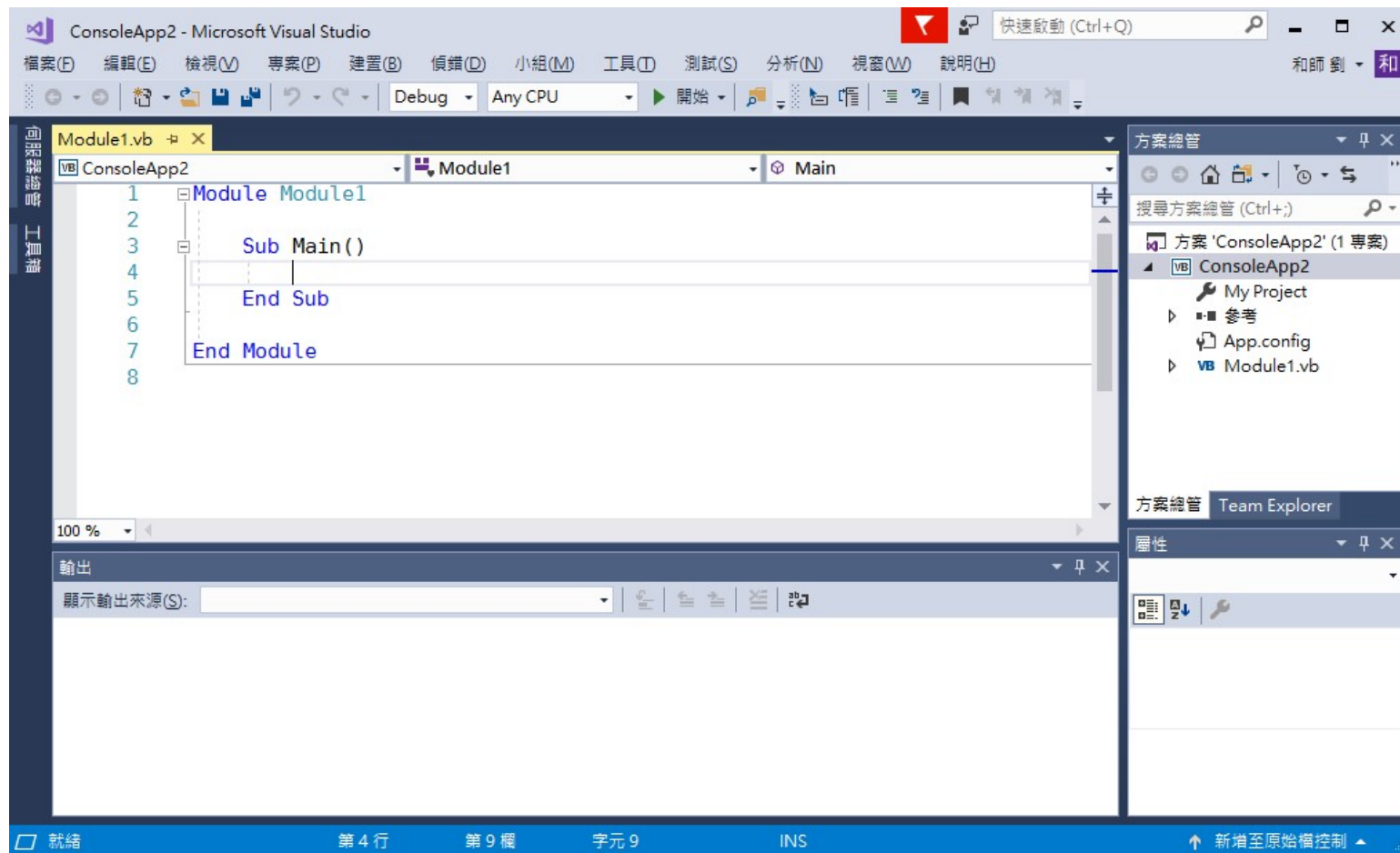
# Visual Basic簡介

## ▶ 點選主控台應用程式(.NET Framework)



# Visual Basic簡介

## ▶ 進入程式撰寫畫面



# Visual Basic簡介

---

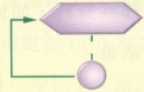
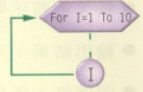

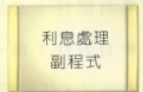

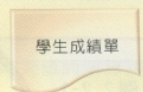


- ▶ 我們會以使用VB2010 Express為主，在Visual Studio 2017也是一樣的，不需修改程式即可執行。
- ▶ 由於再簡單的程式VB環境也會產生許多的檔案和資料夾，所以程式專案儲存位置要管理好。
- ▶ 四技二專統一入學測驗目前仍是以測驗VB的語法為主，所以高職生必須要學會，一般高中學生建議以學習Python或C/C++為主。



# 流程圖(Flow Chart)

- ▶ 解決問題的流程很重要，要先想好：
  - ▶ 1.現在是要解決甚麼問題?
  - ▶ 2.需要輸入什麼?
  - ▶ 3.要獲得什麼輸出?
  - ▶ 4.把解決過程畫(寫)出來，依步驟寫出程式

- ▶ 流程圖是用簡單的圖形來代表執行的步驟，協助我們思考並建立解決步驟

名稱	符號	意義	範例
7.迴圈符號		表示程式迴圈控制變數初值及終值的假設	
8.副程式符號		表示一群程式步驟或流程，用以說明副程式或其他流程的組合	
9.報表符號		表示以列表機印出報表文件	
10.註解符號		表示對某一流程加以註解	



# 流程圖(Flow Chart)

- ▶ 流程圖有許多符號，代表各種意義。
- ▶ 在寫程式前應該都要先畫出流程圖。

名稱	符號	意義	範例
1.起止符號		表示程式的開始或結束	 
2.流程符號		表示流程進行的方向	
3.輸入/輸出符號		表示資料之輸入或結果的輸出	 
4.處理符號		表示執行或處理某些工作	 
5.決策判斷符號		表示對某一個條件做判斷	
6.連接符號		用於：1.轉接到另一頁 2.避免流線交叉 3.避免流線太長	



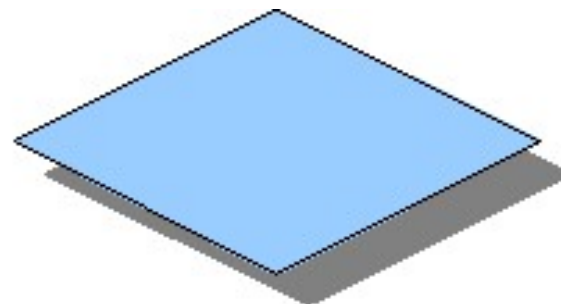
# 流程圖(Flow Chart)

---

- ▶ 對於簡單的問題，其實記住這四個也就夠了



端點



決策



程序步驟



輸出輸入

# 流程圖(Flow Chart)

---

- ▶ 流程圖是你解決問題的想法及過程，沒有標準答案，只要能解決問題就可以
- ▶ 不過效率就有關係了，練習讓你的思考更清晰，步驟更簡潔



## 參考教材

---

- ▶ 我們以李家同教授編寫的「專門為中學生寫的程式語言設計」一書為主，這是一本很簡單的書。
- ▶ 程式依書中範例為主，但改以VB撰寫，習題解答為參考，請要自己親自做出來。



聯經出版

---

加油~~~



# Ch1. 簡單的程式

---

## ▶ 整數加法：

假設我們有兩個整數 $x$ 和 $y$ ，要求輸出 $x+y$ 的結果。

## ▶ 流程圖：

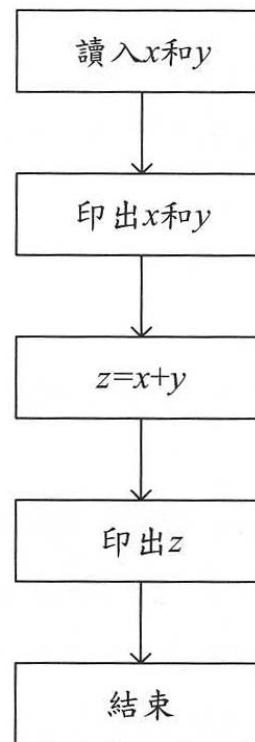


圖 1.1 ▶ 例題 1.1 的流程圖

# Ch1.簡單的程式

---

## ▶ 參考程式：

```
Module Module1
    Sub Main()
        Dim x, y, z As Integer '宣告變數
        Console.Write("Please input the value of variable x: ")
        x = Val(Console.ReadLine())
        Console.Write("Please input the value of variable y: ")
        y = Val(Console.ReadLine())
        Console.WriteLine("x value: " & x) '印出訊息並換行
        Console.WriteLine("y value: " & y)

        z = x + y

        Console.WriteLine("z value: " & z)
        Console.Read() '暫停畫面
    End Sub
End Module
```



# Ch1.簡單的程式

---

## ▶ 說明：

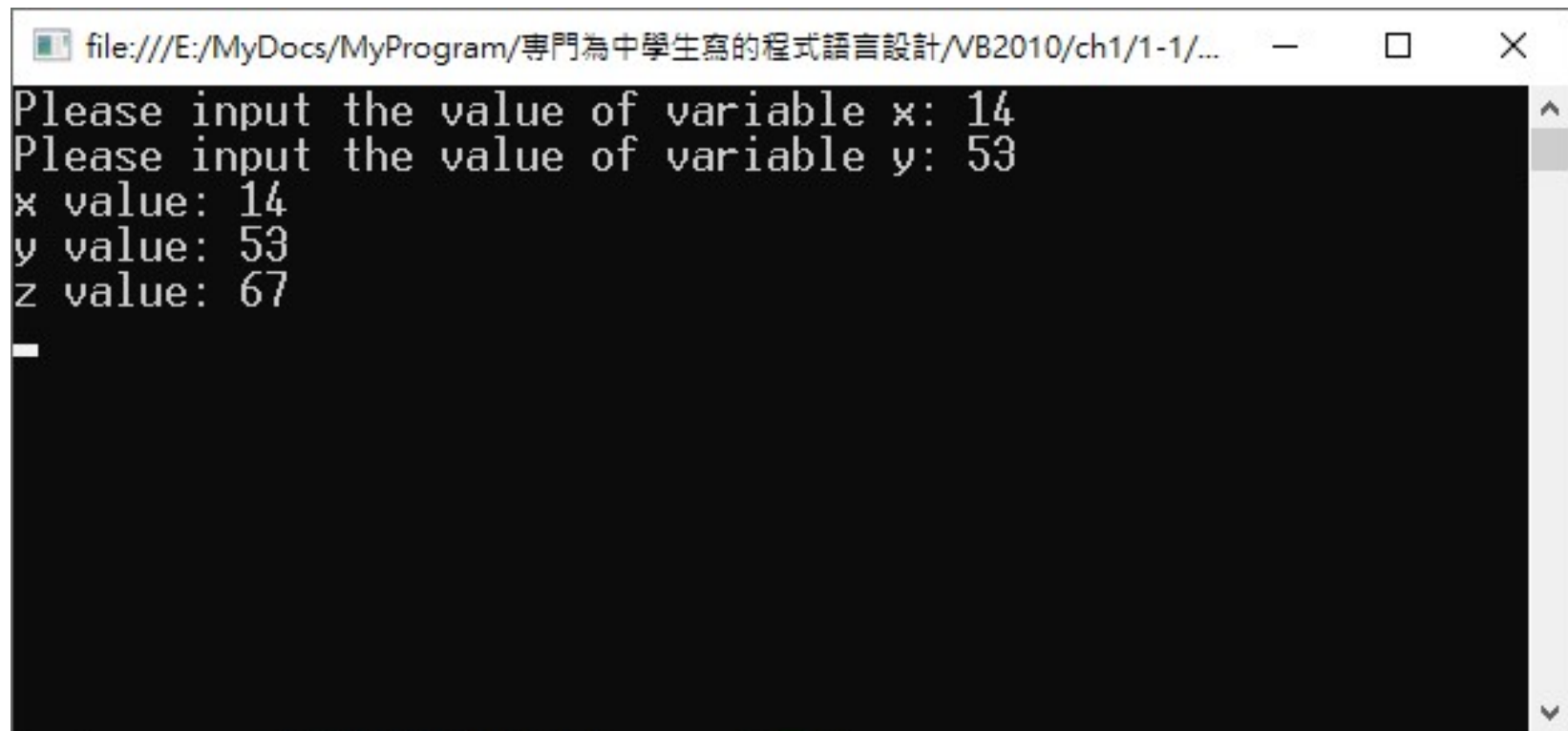
- ▶ 1.深紅色的部分稱“字串”，會忠實地顯示在結果畫面上。
- ▶ 2.x、y、z稱“變數”，會以它們的值呈現，必須先宣告其形態。
- ▶ 3.Console.ReadLine()是請求鍵盤輸入。
- ▶ 4.Console.Write()是印出訊息。
- ▶ 5.Val()是將輸入的字串轉成數值(鍵盤輸入的都視為”字串“)
- ▶ 6.«'»開頭之後的文字為註解，電腦執行時會予以忽略。
- ▶ 7.最後一行Console.Read()命令是要暫停輸出的畫面。
- ▶ 8.執行時空白行會予以忽略，空行是為了程式美觀或易於閱讀。



# Ch1.簡單的程式

---

- ▶ 執行結果：
  - ▶ 會自動叫出DOS視窗，顯示結果。



A screenshot of a DOS window titled "file:///E:/MyDocs/MyProgram/專門為中學生寫的程式語言設計/VB2010/ch1/1-1/...". The window has a black background with white text. The text displays the program's output, which includes prompts for variable values and the resulting values.

```
Please input the value of variable x: 14
Please input the value of variable y: 53
x value: 14
y value: 53
z value: 67
_
```



# Ch1. 簡單的程式

---

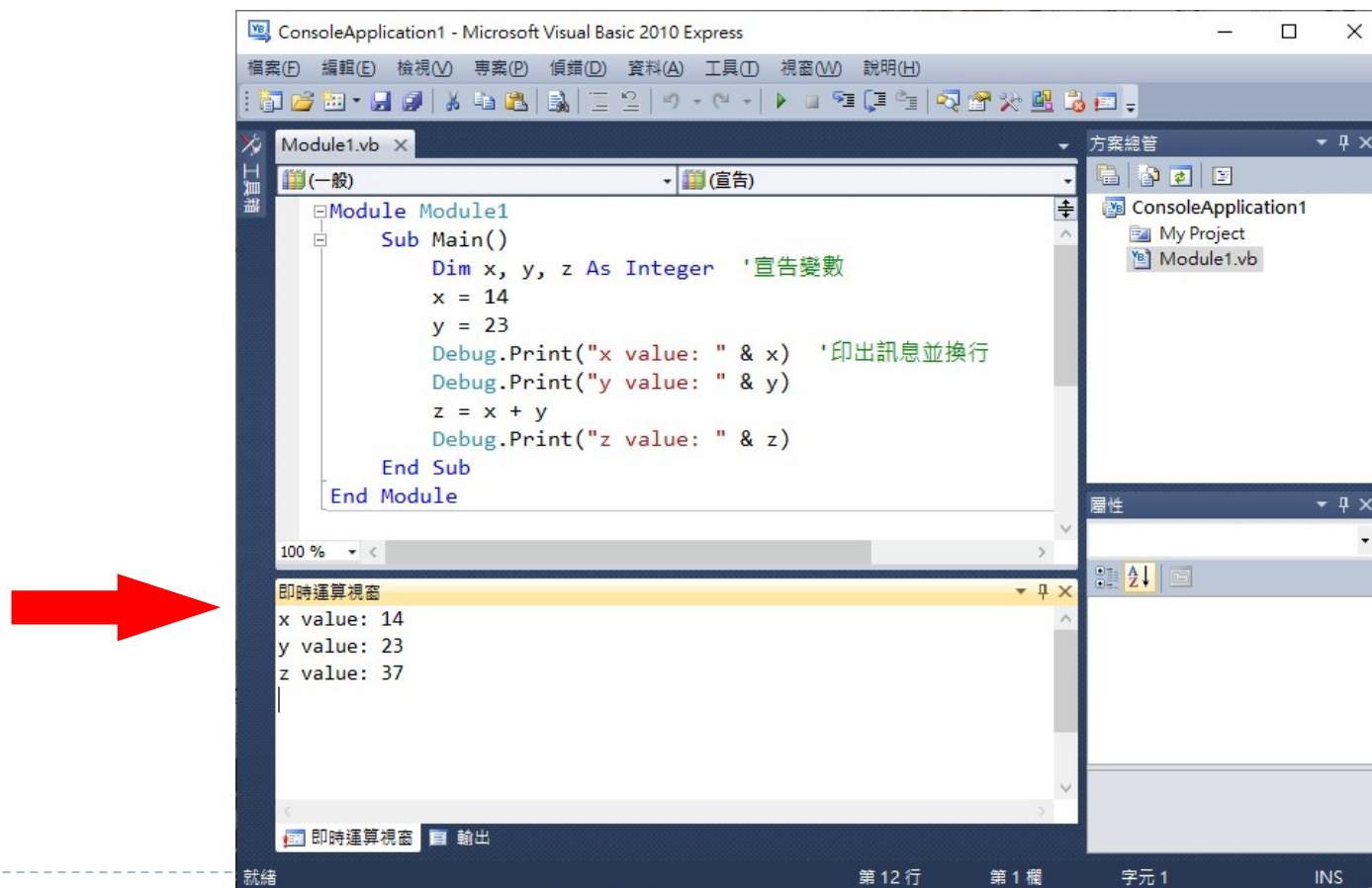
- ▶ 還有一個輸出指令 `Debug.Print()`
- ▶ 它會將結果顯示在”即時運算視窗”，而不是DOS模式下。
- ▶ 一般是在除錯或想特別知道執行中狀況時用，否則還是以 `Console.Write()` 指令為主。
- ▶ 統一入學測驗的考題常會出現這個指令。



# Ch1.簡單的程式

## ▶ 即時運算視窗：

▶ 功能表「偵錯」→「視窗」→「即時運算」。



# Ch1. 簡單的程式

## ▶ 計算 $(x+y)*(u+v)$ :

假設我們有四個整數  $x, y, u$  和  $v$  ,

要求  $(x+y)*(u+v)$

## ▶ 流程圖 :

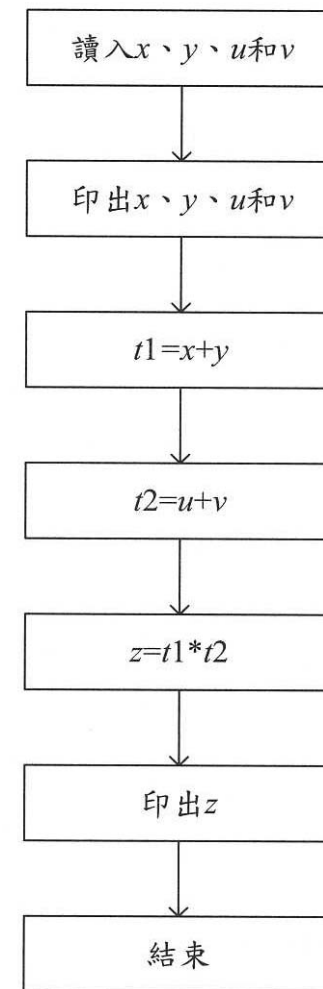


圖 1.3 ▶ 例題 1.3 的流程圖

# Ch1.簡單的程式

## ▶ 參考程式：

```
Module Module1
    Sub Main()
        Dim x, y, u, v, t1, t2, z As Integer '宣告變數
        Console.WriteLine("Please input the value of variable x: ")
        x = Val(Console.ReadLine())
        Console.WriteLine("Please input the value of variable y: ")
        y = Val(Console.ReadLine())
        Console.WriteLine("Please input the value of variable u: ")
        u = Val(Console.ReadLine())
        Console.WriteLine("Please input the value of variable v: ")
        v = Val(Console.ReadLine())
        Console.WriteLine("x value: " & x) '印出訊息並換行
        Console.WriteLine("y value: " & y)
        Console.WriteLine("u value: " & u)
        Console.WriteLine("v value: " & v)
        t1 = x + y
        t2 = u + v
        z = t1 * t2
        Console.WriteLine("z value: " & z)
        Console.Read() '暫停畫面
    End Sub
End Module
```

# Ch1.簡單的程式

---

- ▶ 浮點數字加法：
- ▶ 將變數宣告為Double或Single即可

```
Sub Main()  
    Dim x, y, z As Double '宣告變數  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x value: " & x) '印出訊息並換行  
    Console.WriteLine("y value: " & y)  
    z = x + y  
    Console.WriteLine("z value: " & z)  
    Console.Read() '暫停畫面  
End Sub
```

- ▶ 變數宣告之後其儲存的資料型態就要正確，不能亂存。



# Ch1.簡單的程式

---

## ▶ 輸入文字：

如果要輸入文字，將變數宣告為：

1.Char：字元型態，只能存放一個字。

2.String：字串型態，可以放入一串文字。

```
Dim x As String '宣告變數
Console.Write("Please input your name: ")
x = Console.ReadLine()
Console.WriteLine("How are you? Mr." & x) '印出訊息並換行
Console.Read() '暫停畫面
```

## ▶ 執行結果：



```
file:///C:/Users/Orion Liu/AppDat...
Please input your name: 劉和師
How are you? Mr.劉和師
_
```

# Ch1.簡單的程式

---

- ▶ 計算二次方  $a^2 + b^2$  :

```
Sub Main()  
    Dim a, b, c As Integer '宣告變數  
    Console.Write("請輸入 a = ")  
    a = Val(Console.ReadLine())  
    Console.Write("請輸入 b = ")  
    b = Val(Console.ReadLine())  
  
    c = a^2 + b^2 '計算次方  
  
    Console.WriteLine("c= " & c)  
    Console.Read() '暫停畫面  
End Sub
```

- ▶ 在VB中次方運算使用「^」符號，如計算 $x^y$ ，可寫成 $x^y$



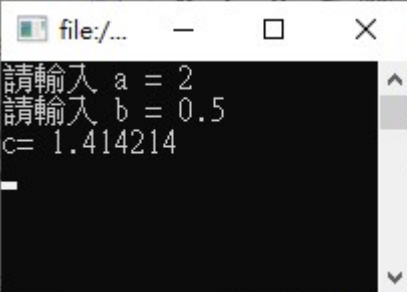
# Ch1.簡單的程式

---

## ▶ 小於1的次方：

- ▶ 計算 $x^y$ ，但 $y$ 為介於0到1之間的浮點數。
- ▶  $y$ 為0.5時就是開平方根號。

```
Sub Main()  
    Dim a, b, c As Single '宣告變數  
    Console.Write("請輸入 a = ")  
    a = Val(Console.ReadLine())  
    Console.Write("請輸入 b = ")  
    b = Val(Console.ReadLine())  
    c = a ^ b  
    Console.WriteLine("c= " & c)  
    Console.Read() '暫停畫面  
End Sub
```



```
file:/...  
請輸入 a = 2  
請輸入 b = 0.5  
c= 1.414214
```



# Ch1.簡單的程式

---

## ▶ 計算 $\log(x)$ ：

```
Sub Main()  
    Dim x, y As Double  
    Console.Write("請輸入 x = ")  
    x = Val(Console.ReadLine())  
  
    y = Math.Log(x)  
  
    Console.WriteLine("Log(" & x & ") =" & y)  
    Console.Read()  
End Sub
```

- ▶ 在這裡 $\log(x)$ 是以 $e = 2.718$ 為底
- ▶ VB內建了許多函數，例如`Math.Log()`函數來求 $\log$ 值，用`Math.Sqrt()`求平方根



# Ch1.簡單的程式

---

- ▶ 計算三角函數  $y = \sin(x)$  :

```
Sub Main()  
    Dim x, y, z As Double  
    Console.Write("請輸入 x = ")  
    x = Val(Console.ReadLine())  
    y = Math.Sin(x)      'x是弧度  
    z = Math.Sin(x * (Math.PI / 180)) '將X轉成角度  
    Console.WriteLine("Sin(" & x & ") =" & y)  
    Console.Read()  
End Sub
```

- ▶ 注意Sin後面跟的數字不是角度，而是弧度，若要計算角度，要轉換一下。
- ▶ VB的  $\pi$  精確到小數點後14位。



## Ch1. 習題

---

- 1-1. 寫一程式，輸入 $a, b, c, d$ ，計算 $\frac{(a+b)}{(c-d)} \times d$ 。先畫出流程圖。

- 1-2. 假設有一組二元一次方程式如下：

$$a_1x + b_1y = c_1 \quad , \quad a_2x + b_2y = c_2$$

此組方程式的解如下：

$$x = \frac{(c_1b_2 - c_2b_1)}{(a_1b_2 - a_2b_1)} \quad , \quad y = \frac{(c_1a_2 - c_2a_1)}{(b_1a_2 - a_1b_2)}$$

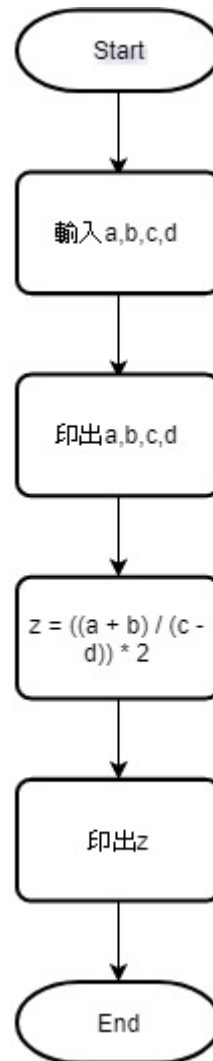
寫一程式，輸入此方程式變數之係數。計算此組方程式的解先畫出流程圖。



# Ch1. 習題

---

## ► 1-1 流程圖：



# Ch1. 習題

---

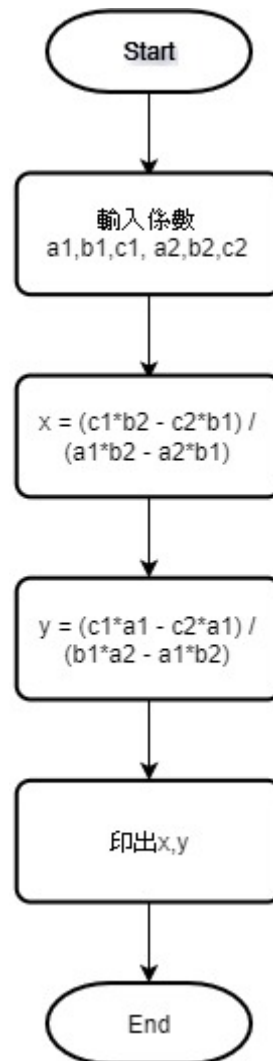
## ► 1-1 參考程式：

```
Sub Main()  
    Dim a, b, c, d, z As Double  
    Console.Write("Please input the value of variable a: ")  
    a = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable b: ")  
    b = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable c: ")  
    c = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable d: ")  
    d = Val(Console.ReadLine())  
    Console.WriteLine("a value: " & a)  
    Console.WriteLine("b value: " & b)  
    Console.WriteLine("c value: " & c)  
    Console.WriteLine("d value: " & d)  
    z = ((a + b) / (c - d)) * 2  
    Console.WriteLine("((a + b) / (c - d)) * 2 = " & z)  
    Console.Read()  
End Sub
```

# Ch1. 習題

---

## ► 1-2 流程圖：



# Ch1. 習題

---

## ► 1-2 參考程式：

```
Sub Main()  
    Dim a1, b1, c1, a2, b2, c2, x, y As Double  
    Console.Write("請輸入係數 a1: ") : a1 = Val(Console.ReadLine())  
    Console.Write("請輸入係數 b1: ") : b1 = Val(Console.ReadLine())  
    Console.Write("請輸入係數 c1: ") : c1 = Val(Console.ReadLine())  
    Console.Write("請輸入係數 a2: ") : a2 = Val(Console.ReadLine())  
    Console.Write("請輸入係數 b2: ") : b2 = Val(Console.ReadLine())  
    Console.Write("請輸入係數 c2: ") : c2 = Val(Console.ReadLine())  
    x = (c1 * b2 - c2 * b1) / (a1 * b2 - a2 * b1)  
    y = (c1 * a1 - c2 * a1) / (b1 * a2 - a1 * b2)  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
    Console.Read()  
End Sub
```

## Ch1. 習題

---

- ▶ 1-3. 輸入a和b，求：

$$c = \frac{a^1 + b^2}{a^2 - b^2}$$

- ▶ 1-4. 輸入a和b，求：

$$c = \sqrt{a^2 + b^2}$$

- ▶ 1-5. 輸入a、b和c，求：

$$c = a - (b + c)(3a - c)$$





# Ch1. 習題

---

## ▶ 1-3 參考程式：

```
Sub Main()  
    Dim a, b, c As Double  
    Console.Write("請輸入a: ")  
    a = Val(Console.ReadLine())  
    Console.Write("請輸入b: ")  
    b = Val(Console.ReadLine())  
  
    c = (a ^ 2 + b ^ 2) / (a ^ 2 - b ^ 2)  
  
    Console.WriteLine("c = " & c)  
    Console.Read()  
End Sub
```



# Ch1. 習題

---

## ▶ 1-4參考程式：

```
Sub Main()  
    Dim a, b, c As Double  
    Console.Write("請輸入a: ")  
    a = Val(Console.ReadLine())  
    Console.Write("請輸入b: ")  
    b = Val(Console.ReadLine())  
  
    c = (a ^ 2 + b ^ 2) ^ 0.5  
  
    Console.WriteLine("c = " & c)  
    Console.Read()  
End Sub
```



# Ch1. 習題

---

## ▶ 1-5 參考程式：

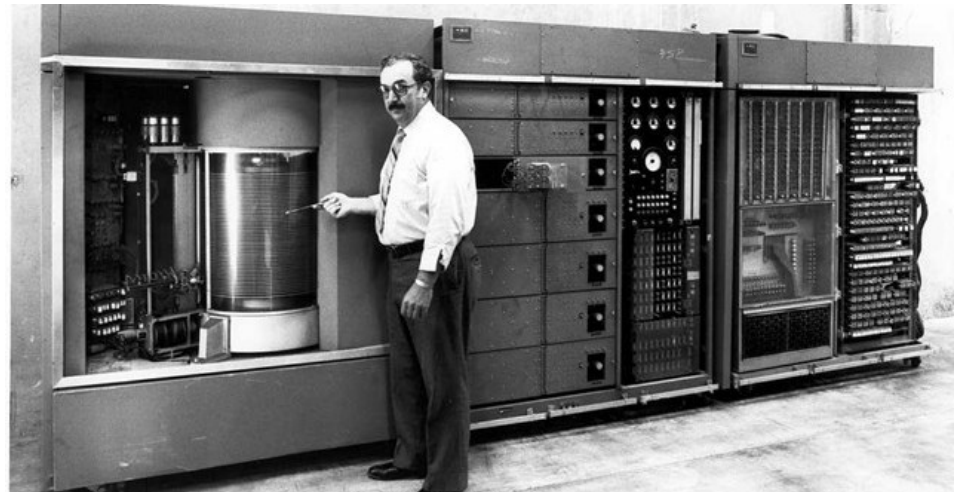
```
Sub Main()  
    Dim a, b, c As Double  
    Console.Write("請輸入a: ") : a = Val(Console.ReadLine())  
    Console.Write("請輸入b: ") : b = Val(Console.ReadLine())  
    Console.Write("請輸入c: ") : c = Val(Console.ReadLine())  
  
    c = a - (b + c) * (3 * a - c)  
  
    Console.WriteLine("c = " & c)  
    Console.Read()  
End Sub
```



## 補充說明：變數、運算式和敘述

---

- ▶ 由於記憶體空間有限以及效率上的考量，我們通常會給變數配置合理的記憶體空間來存放資料
- ▶ VB提供的資料型態：
  - ▶ 整數(Integer)
  - ▶ 浮點數(Float)
  - ▶ 布林(Boolean)
  - ▶ 字元(Character)
  - ▶ 字串(String)
  - ▶ 日期/時間(Date/Time)
  - ▶ 位元組(Byte)
  - ▶ 物件(Object)
  - ▶ 自訂型態(Structure)



# 資料型態一覽表

型態	記憶體	表示範圍
Boolean	Depends on implementing platform	True or False
Byte	1 byte	0 ~ 255 (unsigned)
Char	2 bytes	0 ~ 65535 (unsigned)
Date	8 bytes	0:00:00 (midnight) on January 1, 0001 through 11:59:59 PM on December 31, 9999
Decimal	16 bytes	0 ~ +/-79,228,162,514,264,337,593,543,950,335 (+/-7.9...E+28) <sup>†</sup> with no decimal point; 0 ~ +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.0000000000000000000000000000001 (+/-1E-28) <sup>†</sup>
Double	8 bytes	-1.79769313486231570E+308 ~ -4.94065645841246544E-324 <sup>†</sup> for negative values; 4.94065645841246544E-324 ~ 1.79769313486231570E+308 <sup>†</sup> for positive values

# 資料型態一覽表

型態	記憶體	表示範圍
Integer	4 bytes	-2,147,483,648 ~ 2,147,483,647 (signed)
long	8 bytes	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 (9.2...E+18 <sup>+</sup> ) (signed)
Object (Class)	4 bytes on 32-bit platform 8 bytes on 64-bit platform	Any type can be stored in a variable of type Object
Sbyte	1 byte	-128 ~ 127 (signed)
Short	2 bytes	-32,768 ~ 32,767 (signed)
Single	4 bytes	-3.4028235E+38 ~ -1.401298E-45 <sup>+</sup> for negative values; 1.401298E-45 ~ 3.4028235E+38 <sup>+</sup> for positive values
String (Class)	Depends on implementing platform	0 to approximately 2 billion Unicode characters
UInteger	4 bytes	0 ~ 4,294,967,295 (unsigned)

# 資料型態一覽表

型態	記憶體	表示範圍
Ulong	8 bytes	0 ~ 18,446,744,073,709,551,615 (1.8...E+19 †) (unsigned)
User-Defined (structure)	Depends on implementing platform	Each member of the structure has a range determined by its data type and independent of the ranges of the other members
UShort	2 bytes	0 ~ 65,535 (unsigned)



## 變數宣告範例

---

- ▶ **Dim**          **sum**                      **As**      **Integer**
    - ▶ 宣告    變數名稱                      是                      整數
  - ▶ **Dim**                  **PI**                      **As**      **Double**      =    3.14
    - ▶ 宣告    變數名稱                      是                      倍精度                      初值(可有可無)
  - ▶ **Dim**      **name**      **As**      **Char**      = "A"
  - ▶ **Dim**      **address**      **As**      **String**      =    "花蓮，台灣"
  - ▶ **Dim**      **blnTest**      **As**      **Boolean**
- 
- ▶ 經宣告後程式就知道有這些名稱的記憶體空間可以存放資料
  - ▶ 這些記憶體空間在程式執行結束後便會釋放回系統



## 變數宣告範例

---

▶ **Dim**            **A,        B,        C**            **As**        **Integer**

- ▶ 宣告        這三個變數        都是        整數
- ▶ 一次宣告多個變數時用逗點隔開

▶ **Dim**    **intA,        intB**    **As**    **Integer**

▶ **Dim**    **chrA,        chrB**    **As**    **Char**

- ▶ 有些人習慣在變數命名時加上其型態，這樣看到這個變數就知道它是用來存放何種資料，不易造成混亂

## 變數宣告範例

---

- ▶ 例如：
- ▶ `Dim A As Integer = 0`
- ▶ `Dim B As Char = "0"`
  - ▶ 在電腦中 `0`  **$\neq$**  `"0"`
  - ▶ 數值0是： $00000000_2$       字元0是： $00110000_2$
- ▶ 可以宣告成：
- ▶ `Dim intA As Integer = 0`
- ▶ `Dim chrB As Char = "0"`
  - ▶ 這樣就比較清楚什麼變數放何種資料

## 變數要確定初值

---

- ▶ 變數使用前最好先確定初值，以免發生不確定的錯誤

- ▶ 例如：

```
Dim Sum, A As Integer  
A = Sum + 10
```

- ▶ 因為不知道Sum一開始是多少，所以A也無法確定了
- ▶ 雖然有些程式語言會幫我們把變數初值設為0，但還是自己設定比較保險

- ▶ 應寫成：

```
Dim Sum, A As Integer  
Sum = 0  
A = Sum + 10
```

## 型態轉換

---

- ▶ 數值和字串運算，結果是數值還是字串？
- ▶ 雖然VB會依運算方式自動轉換資料型態，但還是自己確定先轉換比較好
- ▶ 例如：

```
a = 10  
b = "100"  
Console.WriteLine( a + b )  
執行結果： 110
```

```
a = 10  
b = "100"  
Console.WriteLine( a & b )  
執行結果： 10100
```

## 強制型態轉換

---

- ▶ 上例雖然結果沒有錯，但還是自己確定先轉換比較好
- ▶ 應改成：

```
a = 10  
b = CInt("100")  
Console.WriteLine( a + b )  
執行結果： 110
```

```
a = CStr(10)  
b = "100"  
Console.WriteLine( a & b )  
執行結果： 10100
```

- ▶ 雖然結果相同，但可以減少非預期的錯誤

## 自訂型態(Structure)

---

- ▶ 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

# 自訂型態(Structure)

---

► Dim A As Integer

**A**

(放一個整數)



► Dim B As student

**B**

**name**  
(放一個  
字串)

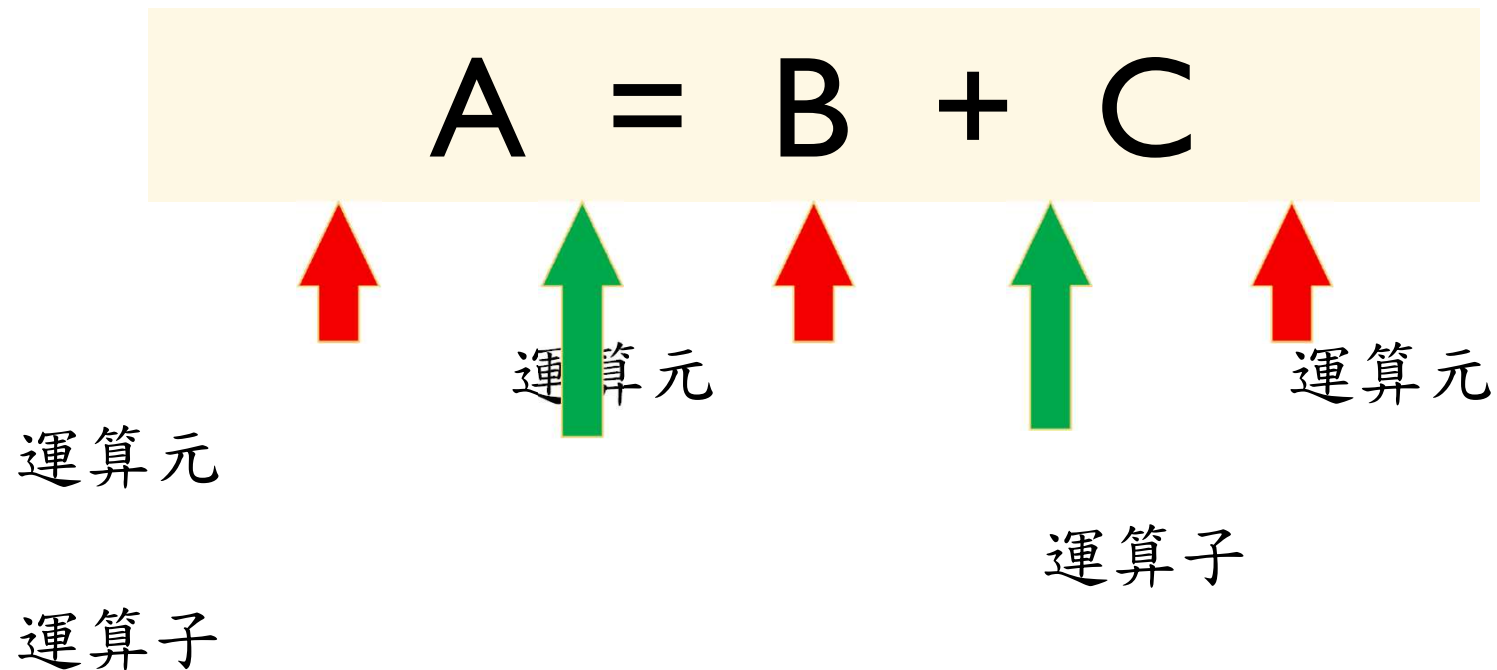
**age**  
(放一  
個數值)

**BMI**  
(放一  
個數值)



## 運算式與運算子

- ▶ 這是一個運算式(Expression)，由運算元(Operand)及運算子(Operator)組成

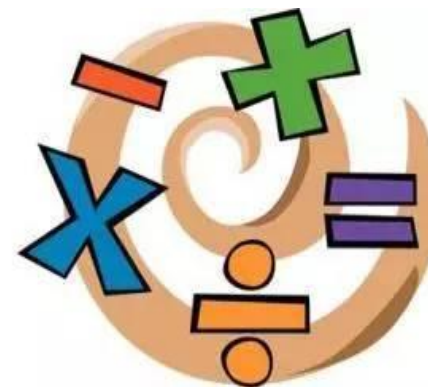




## 運算子類別

---

- ▶ 算術運算子
  - ▶ 指定運算子
  - ▶ 比較/關係運算子
  - ▶ 串接運算子
  - ▶ 邏輯運算子
- 
- ▶ 這些是構成電腦運算式的基本符號



# 算術運算子

表6-3.3 算術運算子

名 稱	符 號	優先次序	數學運算式	VB運算式
次 方	^	1	$X^2$	$X^2$
正 號	+	2	+20	+20
負 號	-	2	-35	-35
乘 號	*	3	$6 \times 3$	$6 * 3$
除 號	/	3	$9 \div 3$	$9 / 3$
整數除法	\	4	無	$6 \backslash 3$
除法餘數	Mod	5	無	$15 \text{ Mod } 4$
加 號	+	6	$9 + 4$	$9 + 4$
減 號	-	6	$9 - 4$	$9 - 4$

## 算術運算子

---

▶ 除法：

▶  $10 / 3 = 3.3333\dots$

▶ 整除：

▶  $10 \setminus 3 = 3$

▶ 餘數：

▶  $10 \text{ MOD } 3 = 1$

$$\begin{array}{r} 3 \overline{) 10} \\ \underline{9} \phantom{0} \\ 1 \end{array}$$

## 算術運算子

---

- ▶  $2^{10}$  要寫成  $2^{\wedge}10$
- ▶ 開根號  $\sqrt{2}$  要寫成  $2^{\wedge}(1/2)$  或  $2^{\wedge}0.5$
- ▶ 開三次方  $\sqrt[3]{2}$  要寫成  $2^{\wedge}(1/3)$
- ▶ 請將下列運算式寫成程式的表示法：

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- ▶ 答案：

$$(-b + (b^2 - 4*a*c)^{0.5}) / (2*a)$$

# 指定運算子

表6-3.7 指派運算子

名 稱	符 號	範 例	相同運算式
指派	=	X = 2	無
相加指派	+=	X += 2	X = X + 2
相減指派	-=	X -= 2	X = X - 2
相乘指派	*=	X *= 2	X = X * 2
相除指派	/=	X /= 2	X = X / 2
整除指派	\=	X \= 2	X = X \ 2
次方指派	^=	X ^= 2	X = X ^ 2
串接指派	&=	X &= "Hello"	X = X & "Hello"

## 比較/關係運算子

表6-3.4 關係運算子

名 稱	符 號	範 例	範例運算結果
等 於	=	6 = 6	True
不等於	<>	4 <> 8	True
小 於	<	8 < 12	True
大 於	>	5 > 6	False
小於或等於	<=	7 <= 5	False
大於或等於	>=	9 >= 5	True

# IS 比較運算

- ▶ 判斷兩者是否為完全相同的物件



- ▶ objX **is** objY 結果為 True
- ▶ objY **is** objZ 結果為 False
- ▶ objX **is** objZ 結果為 False

# LIKE 比較運算

---

- ▶ 是功能強大的字串比較

結果 = 字串 **LIKE** 樣板

- ▶ 字串：欲比對的字串
- ▶ 樣板：由特定字元表示

樣板字元	對應字元
?	是否符合任何單一字元
*	是否符合一個或多個字元
#	是否符合任何數字
(字元集)	是否符合字元集中的內容
(!字元集)	是否符合字元集中的內容



# LIKE 比較運算

---

- ▶ LIKE 舉例：
- ▶ Result = "J" Like "J" → True
- ▶ Result = "J" Like "John" → False
- ▶ Result = "John" Like "J?n" → False
- ▶ Result = "John" Like "J\*n" → True
- ▶ Result = "J" Like "(A-M)" → True
- ▶ Result = "J" Like "(!A-M)" → False

想想看為甚麼？



## 串接運算子

---

- ▶ **&**：可串接不同型態的資料

- ▶ 例如：

**X = "ABC" & 123 & 3.14**

結果為：ABC1233.14

- ▶ **+**：只能串接文字

**X = "ABC" + 123 + 3.14**

結果為：(錯誤訊息)

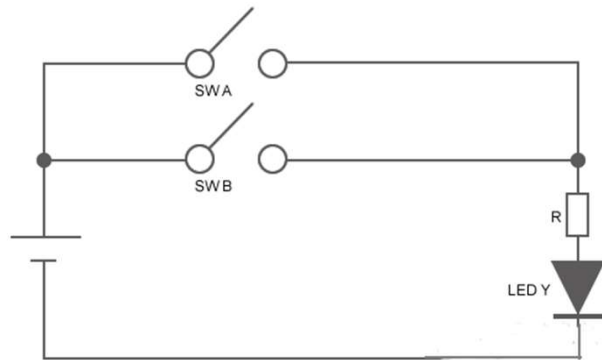
- ▶ 改成：

**X = "ABC" + "123" + "3.14"**

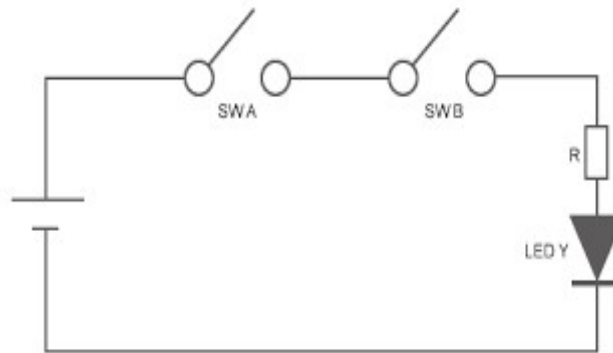
結果為：ABC1233.14

# 邏輯運算子

- ▶ OR的概念：有任一個真則結果為真



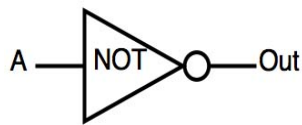
- ▶ AND的概念：有任一個假則結果為假



# 真值表(Truth Table)

- ▶ 邏輯運算子有 NOT、AND、OR、XOR
- ▶ 真值表如下：

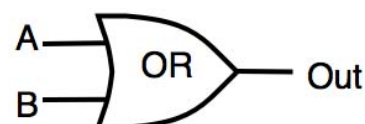
NOT 反相	
A	Out
0	1
1	0



AND 且		
A	B	Out
0	0	0
0	1	0
1	0	0
1	1	1



OR 或		
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	1



XOR 互斥或		
A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0



- ▶ 還有 AndAlso、OrElse，不過很少用

## 運算的優先順序

運算子	說明
()	括符
^	次方
+、-	正負號
*、/	乘除
\	整除
MOD	餘數
+、-	加減
&	字串串接
=、<>、<、>、<=、>=	關係比較
NOT	否
AND	且
OR	或
XOR	互斥或

休息一下~

---

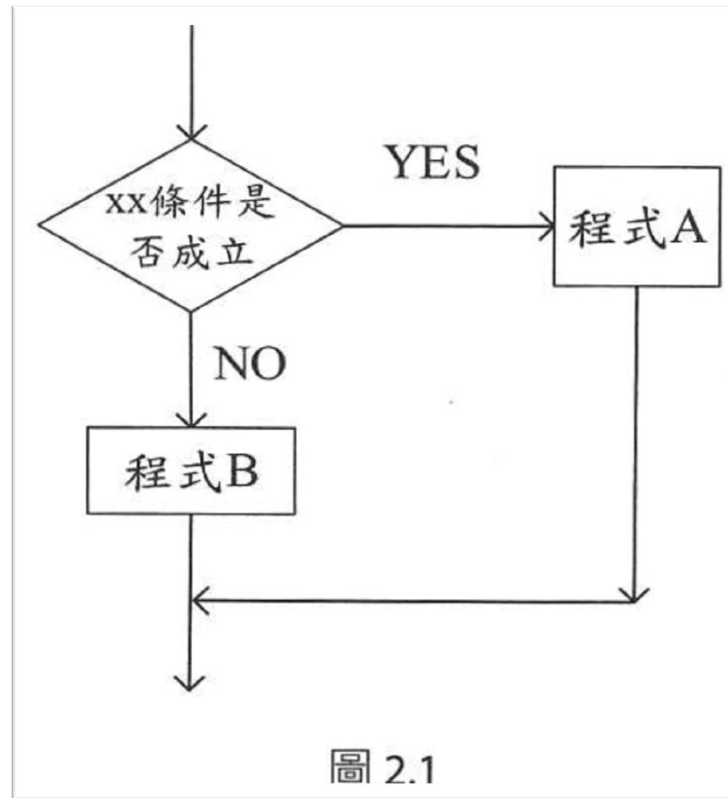


程式設計師的終極形態



## Ch2.有If指令的程式

- ▶ If是：「如果怎樣怎樣，你就該如何如何」
- ▶ 圖2.1的意義是：如果某某條件成立，執行程式A，否則執行程式B。



## Ch2.有If指令的程式

- ▶ 讀入兩個數 $x$ 和 $y$ ，選出大的那一個。
- ▶ 流程圖：

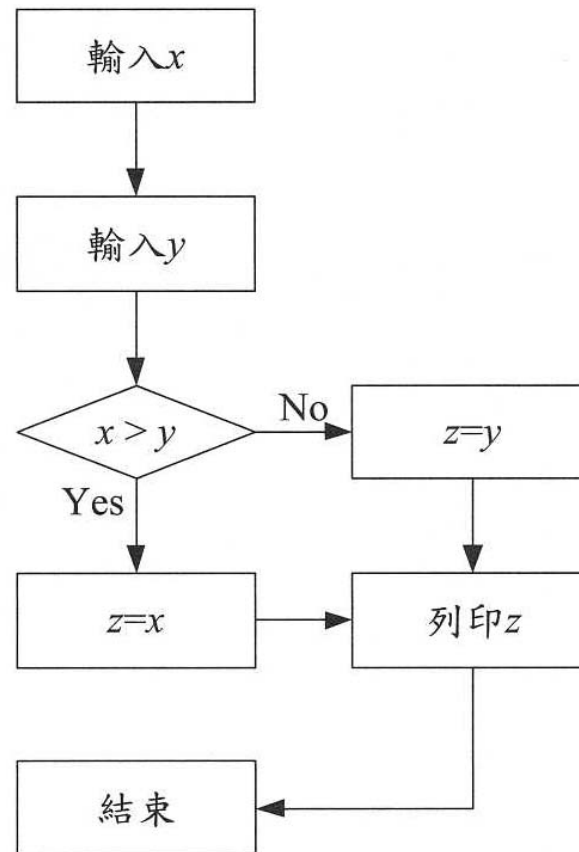


圖 2.2 ▶ 例題 2.1 的流程圖



## Ch2.有If指令的程式

---

### ▶ 參考程式：

```
Sub Main()  
    Dim x, y, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
  
    If x > y Then  
        z = x  
    Else  
        z = y  
    End If  
  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2.有If指令的程式

- ▶ 讀入 $x$ 和 $y$ ，如果 $x > y$ ，則回傳 $x - y$ ，否則回傳 $y - x$ 。
- ▶ 流程圖：

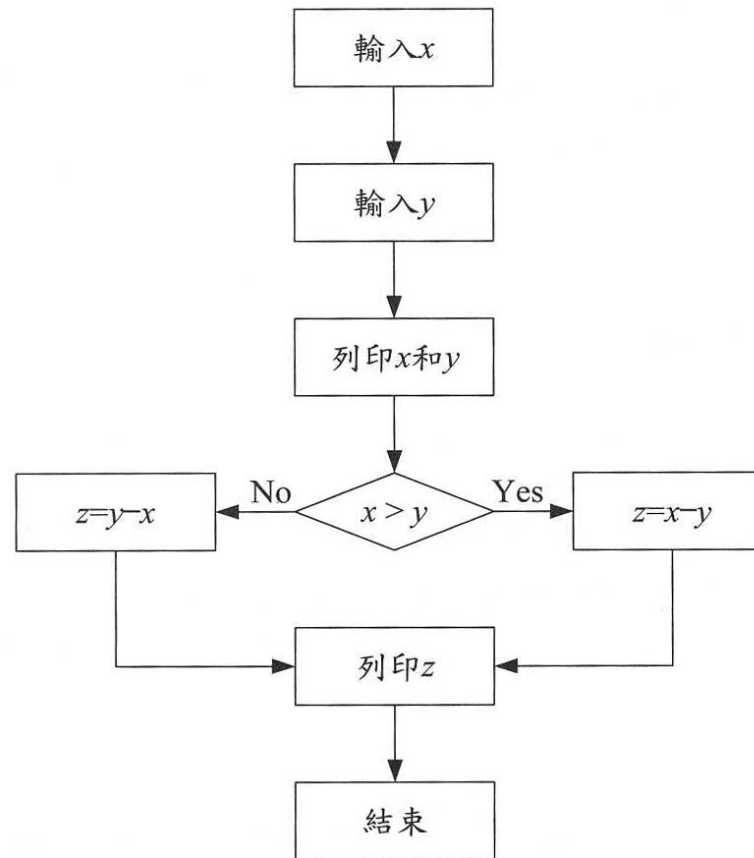


圖 2.3 ▶ 例題 2.2 的流程圖

## Ch2.有If指令的程式

---

### ▶ 參考程式：

```
Sub Main()  
    Dim x, y, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
  
    If x > y Then  
        z = x - y  
    Else  
        z = y - x  
    End If  
  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2. 有If指令的程式

---

- ▶ 讀入一數字，如果是負數，就將它轉成正數。
- ▶ 流程圖：

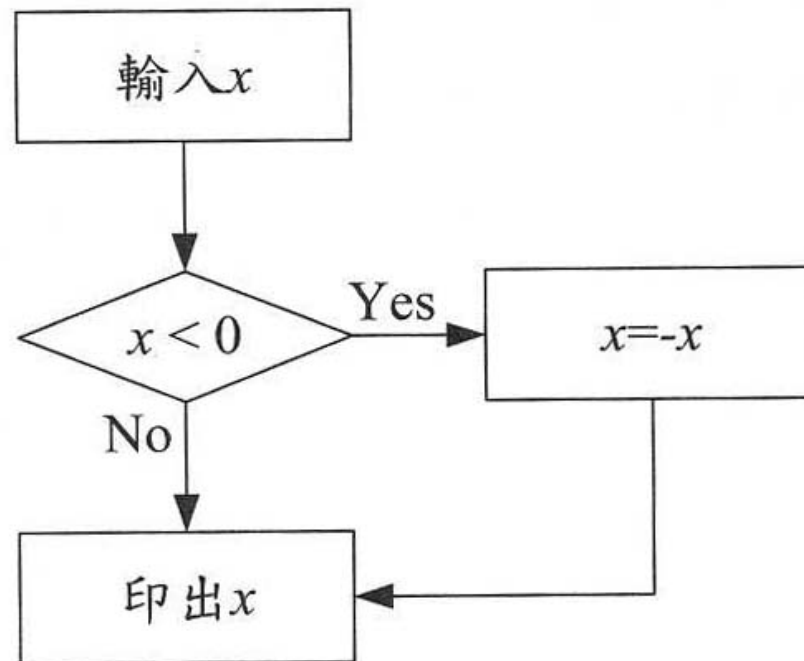


圖 2.4 ▶ 例題 2.3 的流程圖

## Ch2.有If指令的程式

---

### ▶ 參考程式：

```
Sub Main()  
    Dim x As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
  
    If x < 0 Then  
        x = -x  
    End If  
  
    Console.WriteLine("x = " & x)  
    Console.Read()  
End Sub
```



## Ch2.有If指令的程式

- ▶ 輸入一個學生的分數，輸出該成績級別，分數級別如下：

90分以上 → A

80~89分 → B

70~79分 → C

60~69分 → D

60分以下 → F

- ▶ 流程圖：

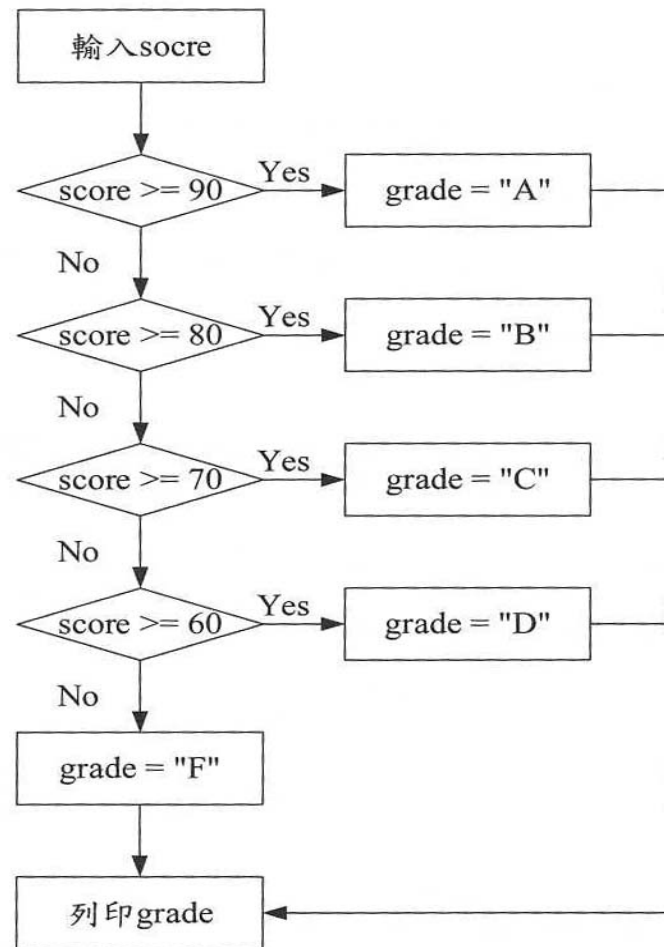


圖 2.5 ▶ 例題 2.4 的流程圖

## Ch2.有If指令的程式

---

### ▶ 參考程式一：

```
Sub Main()  
    Dim score As Integer  
    Dim grade As Char  
    Console.WriteLine("輸入學生成績，將會輸出對應的成績級別")  
    Console.Write("輸入成績(0~100): ")  
    score = Val(Console.ReadLine())  
    If score >= 90 Then  
        grade = "A"  
    ElseIf score >= 80 Then  
        grade = "B"  
    ElseIf score >= 70 Then  
        grade = "C"  
    ElseIf score >= 60 Then  
        grade = "D"  
    Else  
        grade = "F"  
    End If  
    Console.WriteLine("此成績級別為 " & grade)  
    Console.Read()  
End Sub
```

## Ch2.有If指令的程式

### ▶ 參考程式二：

```
Sub Main()  
    Dim score As Integer  
    Dim grade As Char  
    Console.WriteLine("輸入學生成績，將會輸出對應的成績級別")  
    Console.Write("輸入成績(0~100): ")  
    score = Val(Console.ReadLine())  
    Select Case score  
        Case Is >= 90  
            grade = "A"  
        Case Is >= 80  
            grade = "B"  
        Case Is >= 70  
            grade = "C"  
        Case Is >= 60  
            grade = "D"  
        Case Else  
            grade = "F"  
    End Select  
    Console.WriteLine("此成績級別為 " & grade)  
    Console.Read()  
End Sub
```



## Ch2.有If指令的程式

---

- ▶ 某商店在大特價，根據購買的商品總金額決定打折的折扣，總金額越高折扣越多，折扣表如下：

未滿1000 → 無折扣

1000~4999 → 9折

5000~9999 → 8折

高於10000 → 7折

小數點四捨五入，印出折扣後的金額。



## Ch2.有If指令的程式

### ► 流程圖：

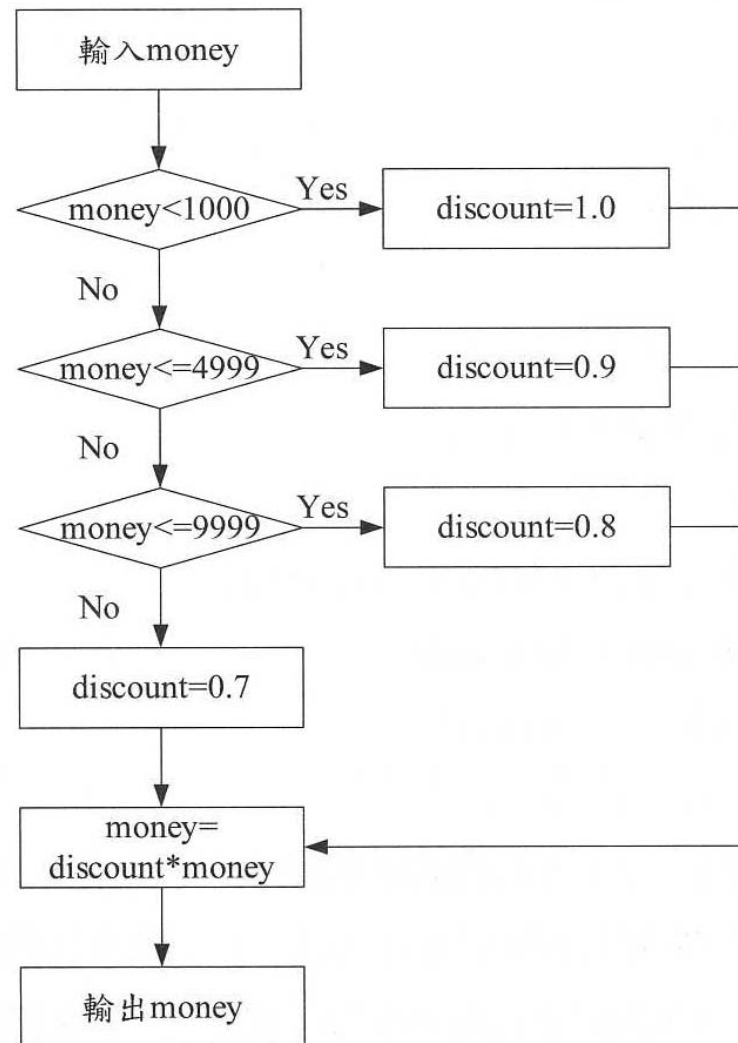


圖 2.6 ► 例題 2.5 的流程圖

## Ch2.有If指令的程式

---

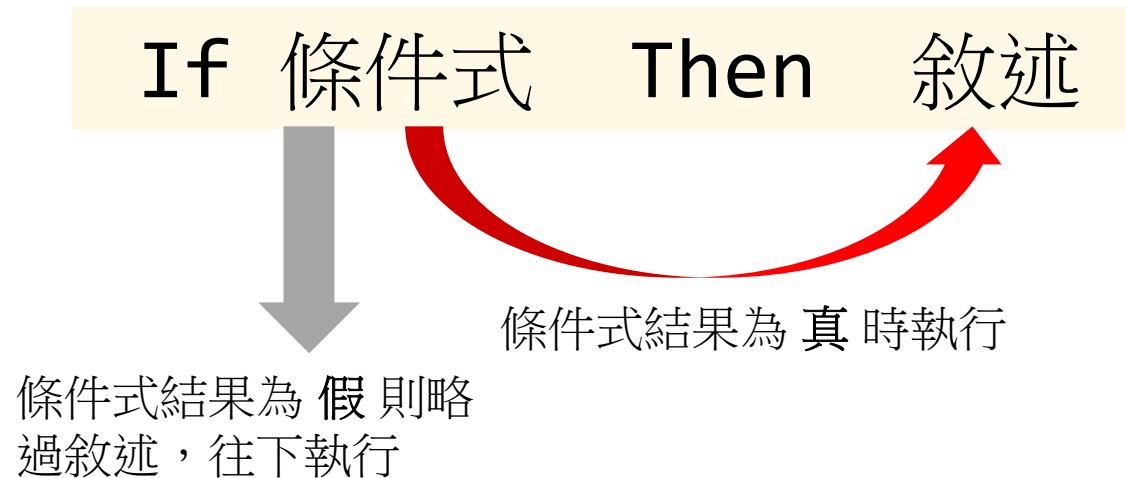
### ▶ 參考程式：

```
Sub Main()  
    Dim money As Integer  
    Dim discount As Single  
    Console.WriteLine("輸入總金額，將會依據金額打折，並輸出打折後的金額")  
    Console.Write("請輸入購買總金額：")  
    money = Val(Console.ReadLine())  
    Select Case money  
        Case Is < 1000  
            Console.WriteLine("抱歉，沒有折扣唷!") : discount = 1.0  
        Case Is < 4999  
            Console.WriteLine("享有9折優惠唷!") : discount = 0.9  
        Case Is < 9999  
            Console.WriteLine("享有8折優惠唷!") : discount = 0.8  
        Case Else  
            Console.WriteLine("享有7折優惠唷!") : discount = 0.7  
    End Select  
    money = Math.Round(money * discount, 0)  
    Console.WriteLine("打折後總金額為：" & money)  
    Console.Read()  
End Sub
```

# If... Then指令

---

- ▶ 單行的If：
- ▶ 語法：

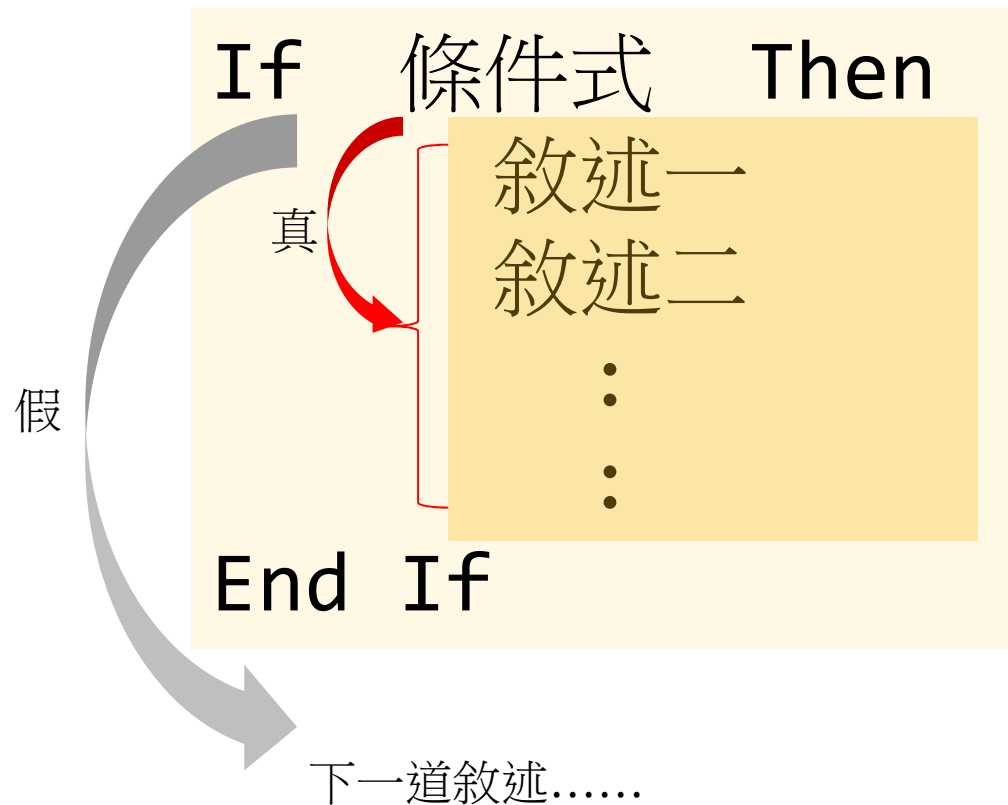


- ▶ 例： **If A > B Then A = 10**

## If... Then... End If 指令

---

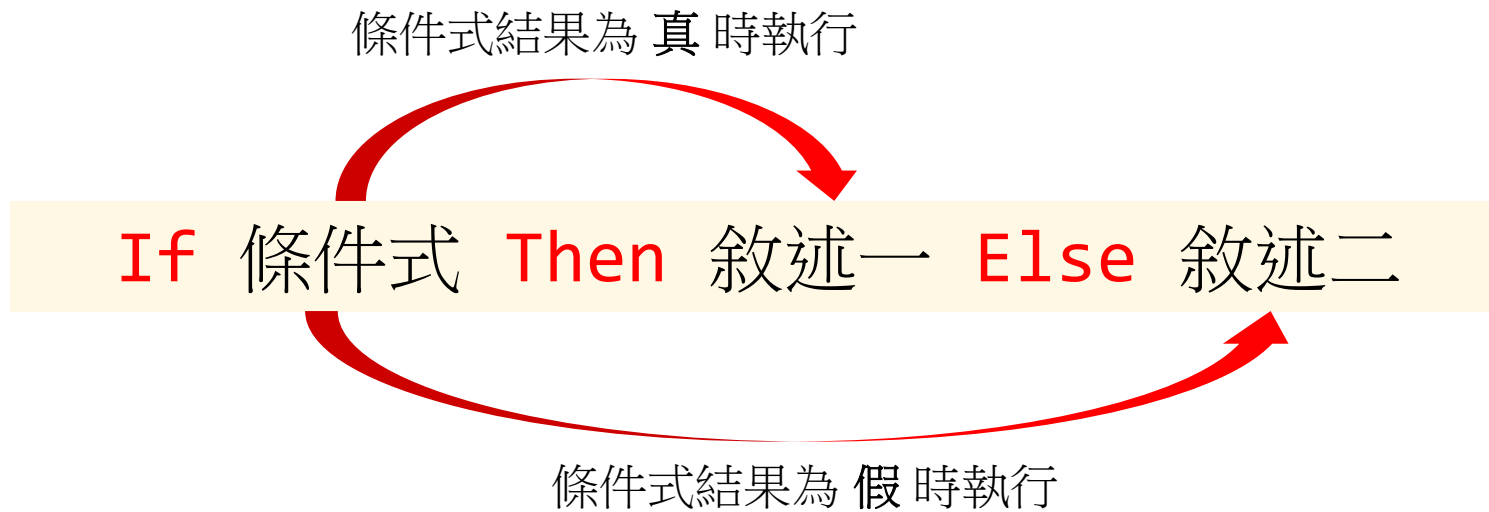
- ▶ 如果條件成立需要執行多行敘述，則可以寫成：



# If... Then... Else 指令

---

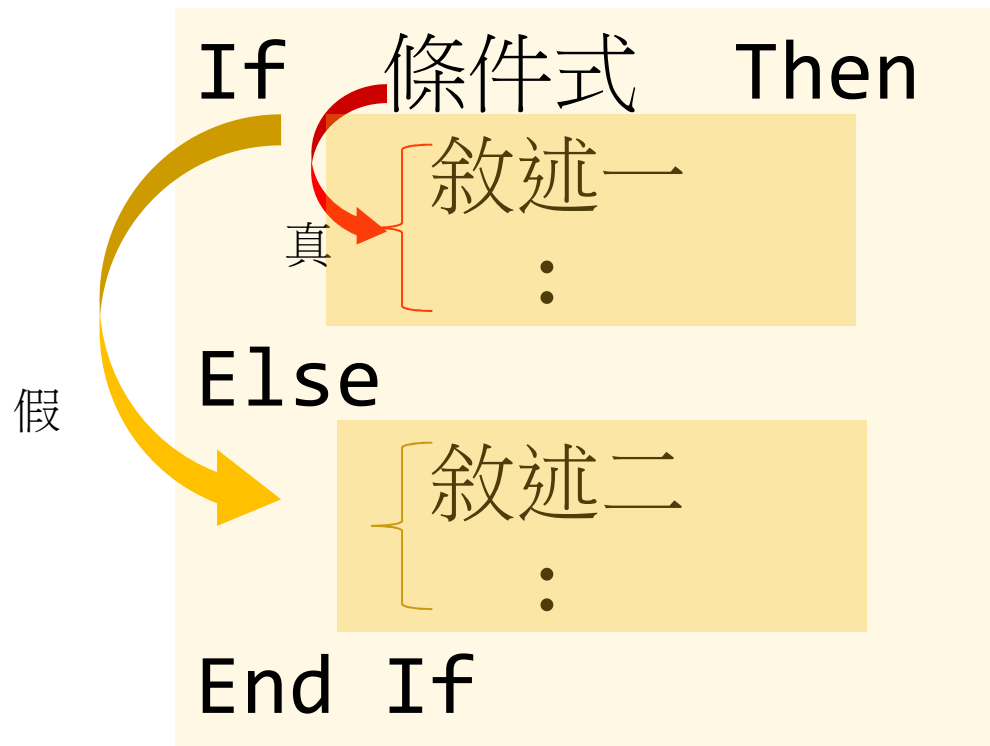
- ▶ 如果... 那就... 否則...
- ▶ 條件式成立或不成立都有敘述要做
- ▶ 語法：



# If... Then... Else 指令

---

- ▶ 多行式寫法：




## IIF( )條件函數

---

- ▶ 當IF的判斷很單純時，可以使用此方式
- ▶ 語法：

變數 = IIF(條件運算式，設定值1，設定值2)



條件運算式為真

條件運算式為假



# If...Then...ElseIf...

- ▶ 當有多重的判斷時，可以這樣用

```
If ... Then
  敘述一
Else
  If ... Then
    敘述二
  Else
    If ... Then
      敘述三
    Else
      敘述四
    End If
  End If
End If
```



```
If ... Then
  敘述一
ElseIf ... Then
  敘述二
ElseIf ... Then
  敘述三
Else
  敘述四
End If
```

有沒有簡化很多呢?

# Select Case 判斷式

---

- ▶ 當需要多重判斷時，還有比If ... ElseIf ... 更簡潔的方式，就是用Select Case，語法：

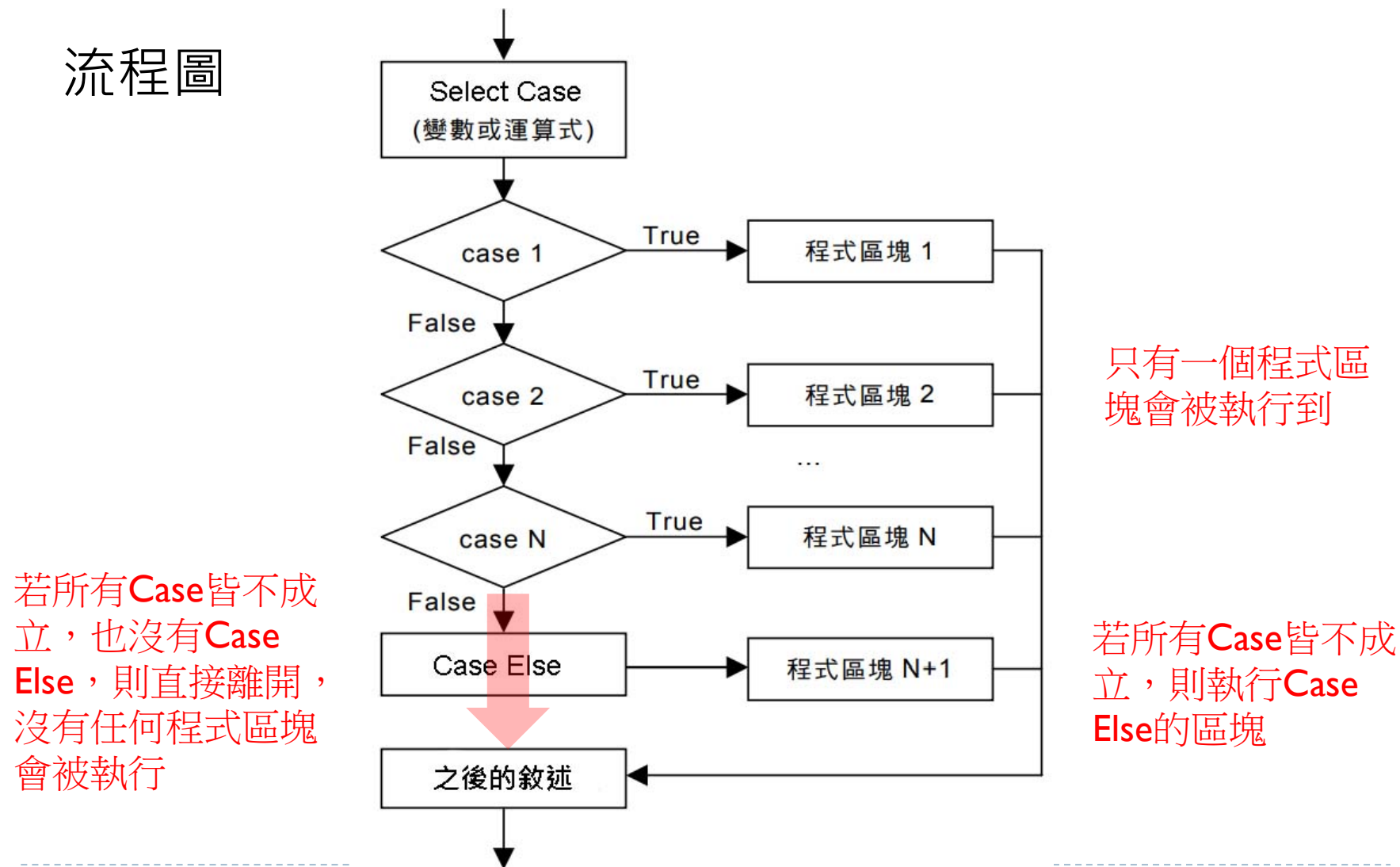
```
Select Case 變數  
    Case 條件式一  
        敘述一  
    Case 條件式二  
        敘述二  
    :  
    Case Else  
        敘述  
End Select
```

可以有很多Case  
第一個符合條件的  
Case 會被執行，之  
後的都忽略

Case Else可省略

# Select Case 判斷式

## 流程圖



## Select Case 判斷式

---

- ▶ Case可接受的運算式：

Case運算式	說明
Case 100	直接寫出數值
Case "ABC"	直接寫出字元或字串
Case 1 To 5	是否介於某一範圍
Case 1, 2, 3	個別列舉
Case Is > 100	是否大或小於某一值

## Ch2. 習題

---

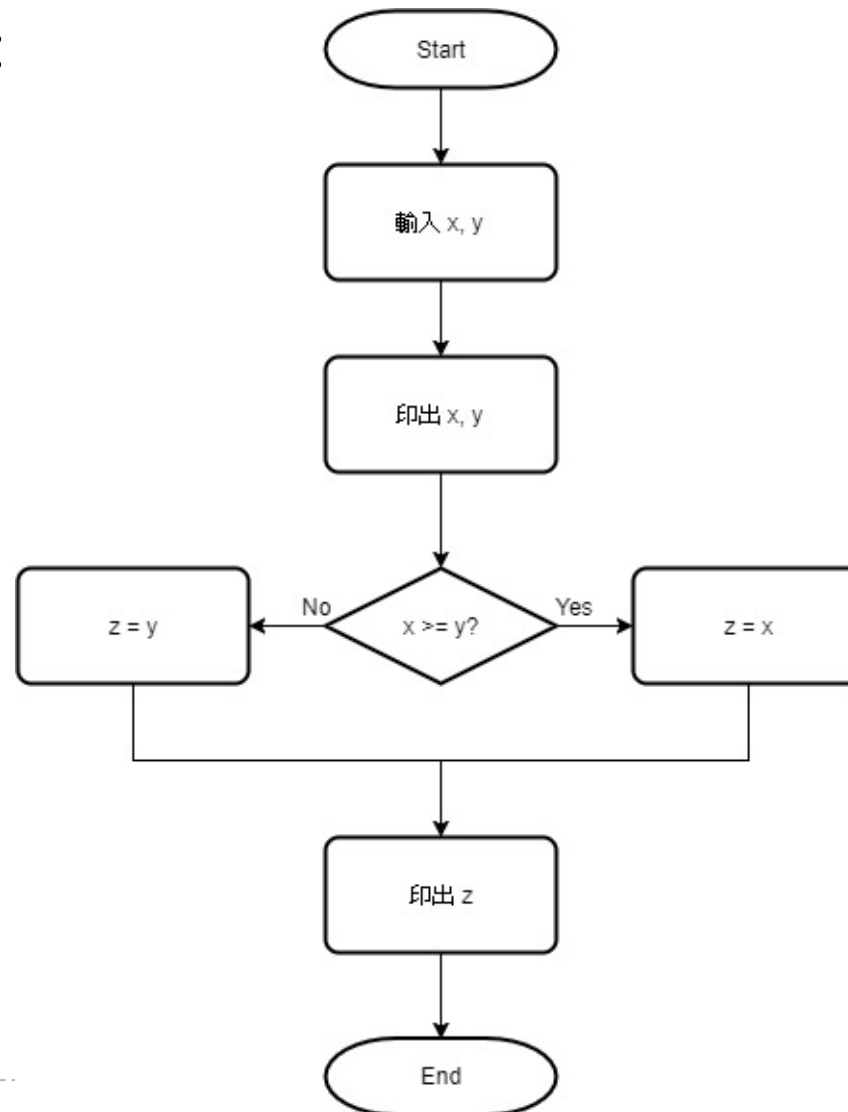
- ▶ 以下各題均需先畫出流程圖。
- ▶ 2-1. 寫一程式，輸入 $x$ 和 $y$ ，如果 $x \geq y$ ，則列印 $x$ ，否則列印 $y$
- ▶ 2-2. 寫一程式，輸入 $x$ 和 $y$ ，如果 $x$ 和 $y$ 都是正數，令 $z=1$ ，如兩者均為負數，令 $z=-1$ ，否則令 $z=0$
- ▶ 2-3. 寫一程式，輸入 $x$ 、 $y$ 、 $u$ 、 $v$ ，如果 $(x+y) > (u+v)$ ，則令 $z=x-y$ ，否則令 $z=u-v$



## Ch2. 習題

---

### ► 2-1 流程圖：



## Ch2. 習題

---

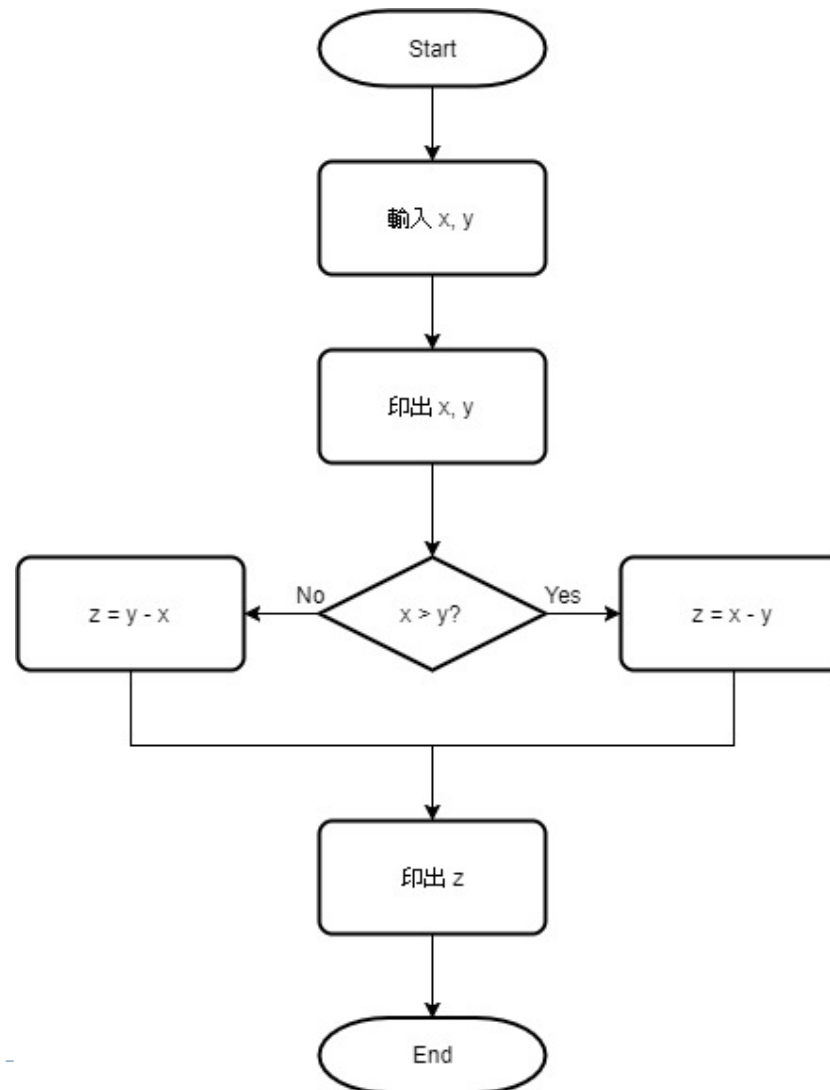
### ► 2-1 參考程式：

```
Sub Main()  
    Dim x, y, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
  
    If x >= y Then  
        z = x  
    Else  
        z = y  
    End If  
  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2. 習題

---

### ► 2-2 流程圖：





## Ch2. 習題

---

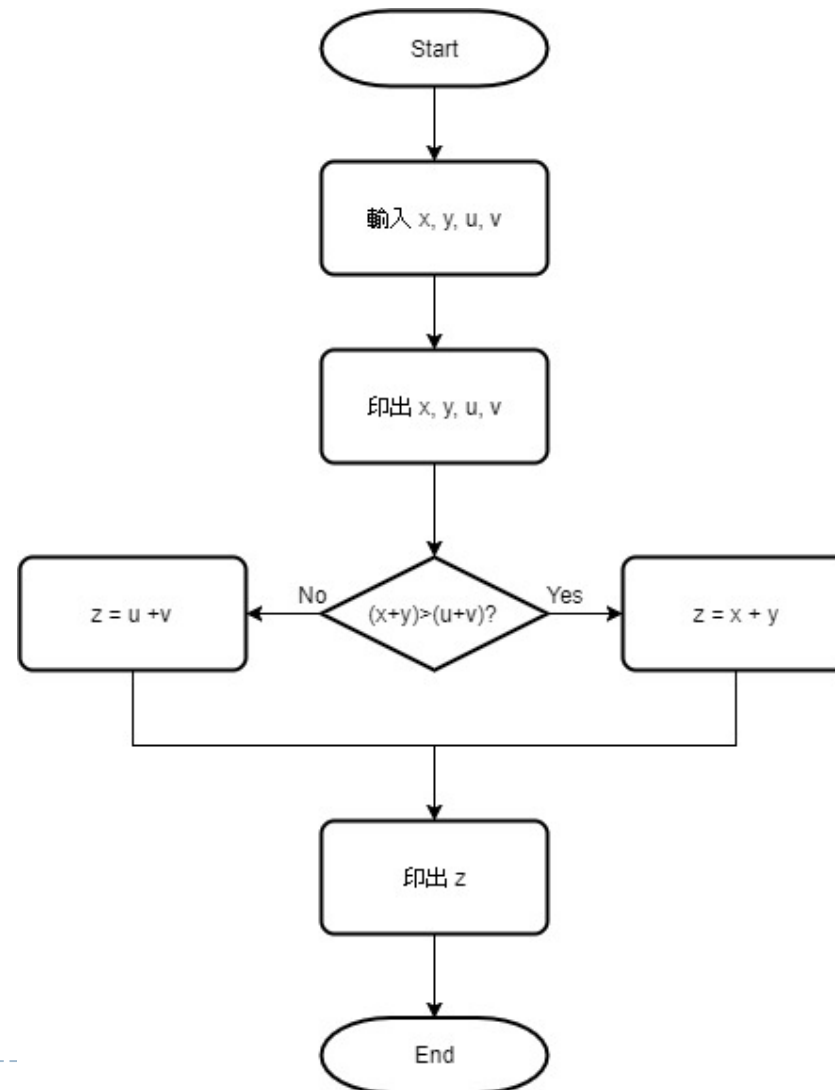
### ► 2-2 參考程式：

```
Sub Main()  
    Dim x, y, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
    If x > 0 And y > 0 Then  
        z = 1  
    ElseIf x < 0 And y < 0 Then  
        z = -1  
    Else  
        z = 0  
    End If  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2. 習題

---

### ► 2-3 流程圖：



## Ch2. 習題

---

### ► 2-3 參考程式：

```
Sub Main()  
    Dim x, y, u, v, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable u: ")  
    u = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable v: ")  
    v = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x) : Console.WriteLine("y = " & y)  
    Console.WriteLine("u = " & u) : Console.WriteLine("v = " & v)  
    If (x + y) > (u + v) Then  
        z = x + y  
    Else  
        z = u + v  
    End If  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2. 習題

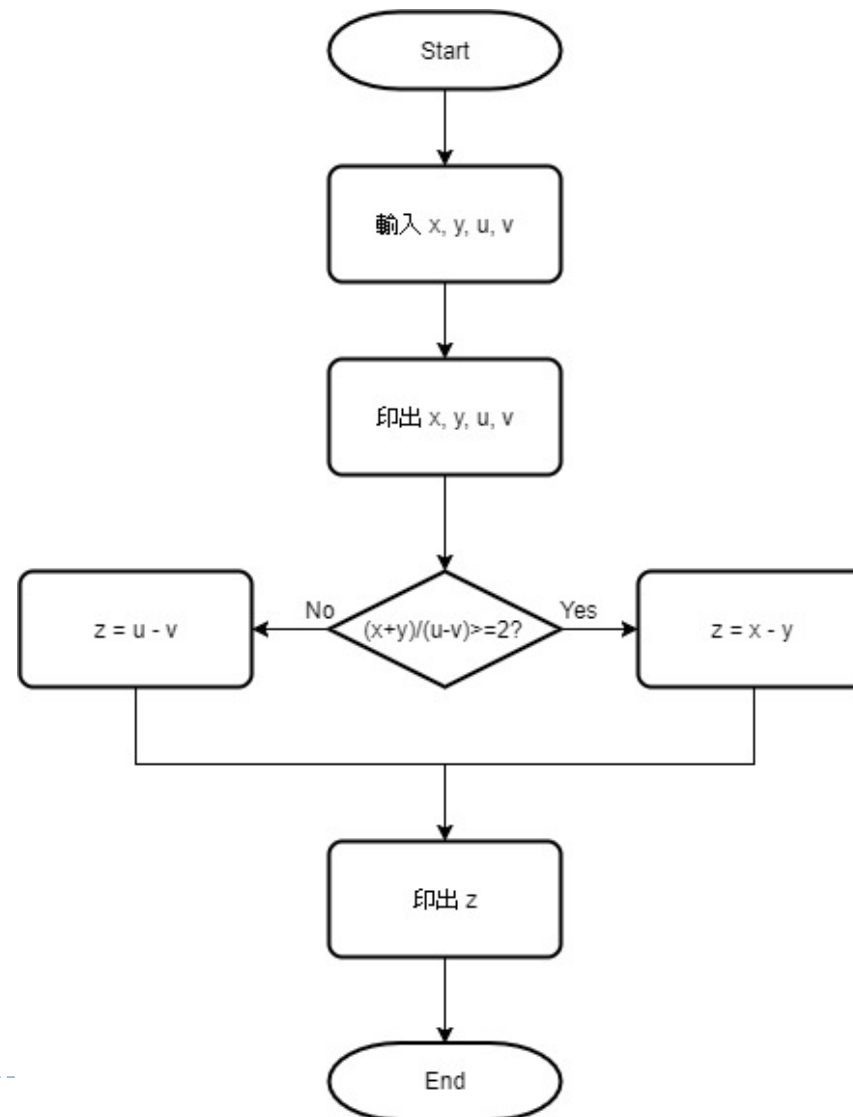
---

- ▶ 2-4. 寫一程式，輸入 $x$ 、 $y$ 、 $u$ 、 $v$ ，如果 $\frac{x+y}{u-v} \geq 2$ ，令 $z=x-y$ ，否則令 $z=u-v$ 。
- ▶ 2-5. 寫一程式，輸入 $x$ 和 $y$ ，如果 $x \geq y$ ，令 $z=x^2$ ，否則令 $z=y^2$



## Ch2. 習題

### ► 2-4 流程圖：



## Ch2. 習題

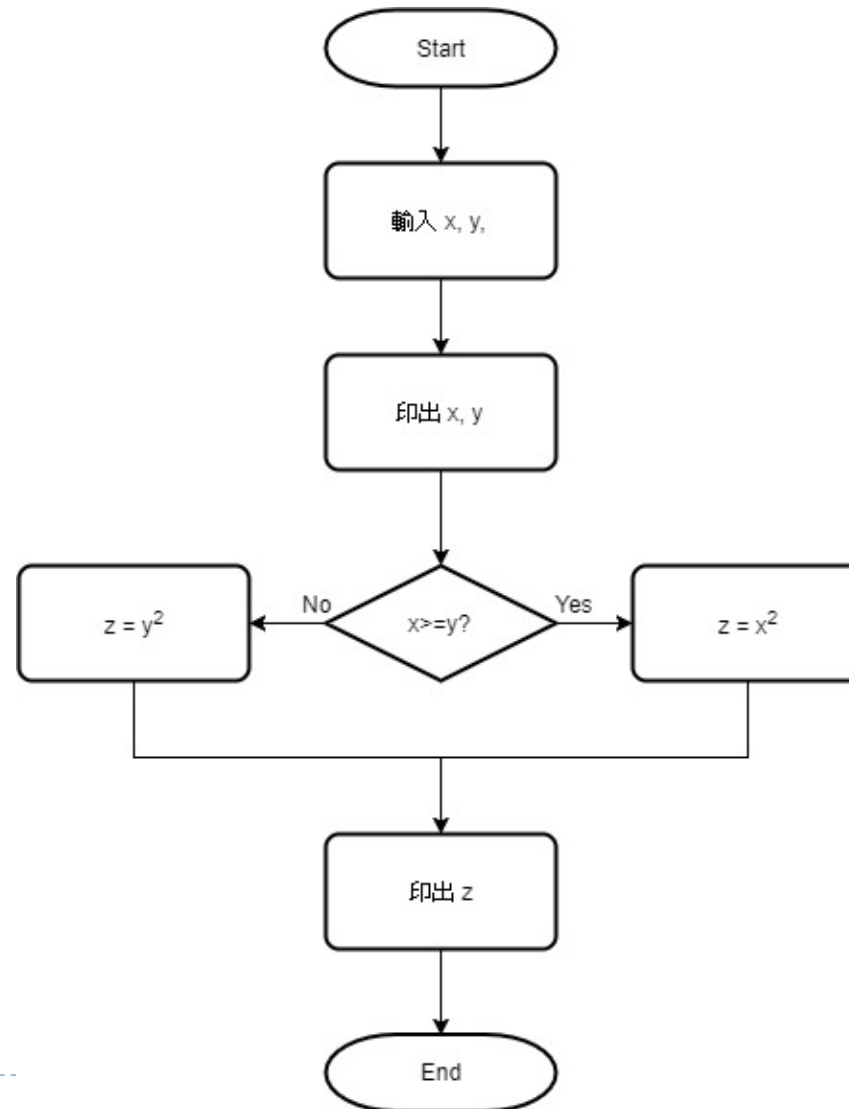
---

### ► 2-4 參考程式：

```
Sub Main()  
    Dim x, y, u, v, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable u: ")  
    u = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable v: ")  
    v = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x) : Console.WriteLine("y = " & y)  
    Console.WriteLine("u = " & u) : Console.WriteLine("v = " & v)  
    If (x + y) / (u - v) Then  
        z = x - y  
    Else  
        z = u - v  
    End If  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```

## Ch2. 習題

### ► 2-5 流程圖



## Ch2. 習題

---

### ► 2-5 參考程式：

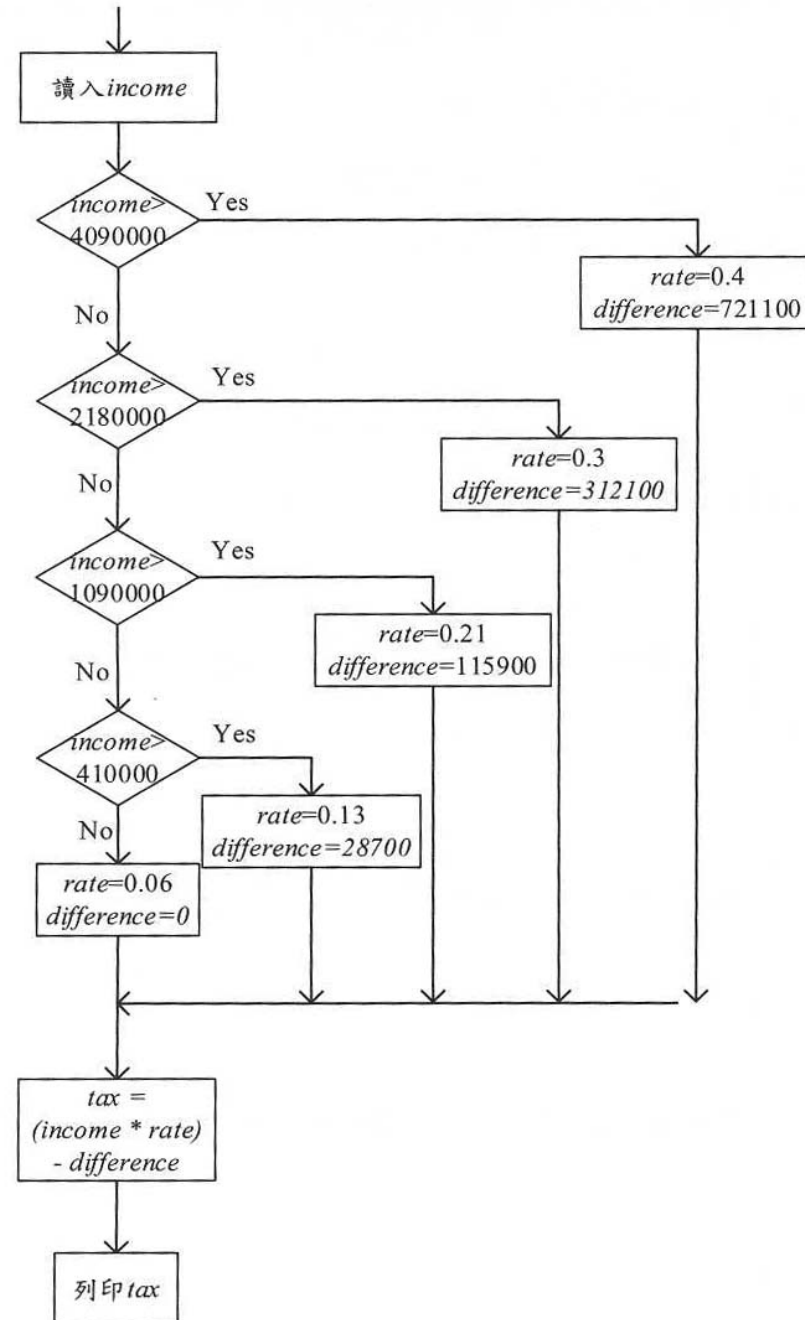
```
Sub Main()  
    Dim x, y, z As Integer  
    Console.Write("Please input the value of variable x: ")  
    x = Val(Console.ReadLine())  
    Console.Write("Please input the value of variable y: ")  
    y = Val(Console.ReadLine())  
    Console.WriteLine("x = " & x)  
    Console.WriteLine("y = " & y)  
    If x >= y Then  
        z = x ^ 2  
    Else  
        z = y ^ 2  
    End If  
    Console.WriteLine("z = " & z)  
    Console.Read()  
End Sub
```





## Ch2. 習題

- 2-6. 依照以下的流程圖，  
寫一程式。



## Ch2. 習題

---

### ► 2-6 參考程式：

```
Sub Main()  
    Dim income, difference As Integer  
    Dim rate, tax As Single  
    Console.WriteLine("Please input your income: ")  
    income = Val(Console.ReadLine())  
    Select Case income  
        Case income > 4090000  
            rate = 0.4 : difference = 721100  
        Case income > 2180000  
            rate = 0.3 : difference = 312100  
        Case income > 1090000  
            rate = 0.21 : difference = 115900  
        Case income > 410000  
            rate = 0.13 : difference = 28700  
        Case Else  
            rate = 0.06 : difference = 0  
    End Select  
    tax = (income * rate) - difference  
    Console.WriteLine("tax = " & tax)  
    Console.Read()  
End Sub
```

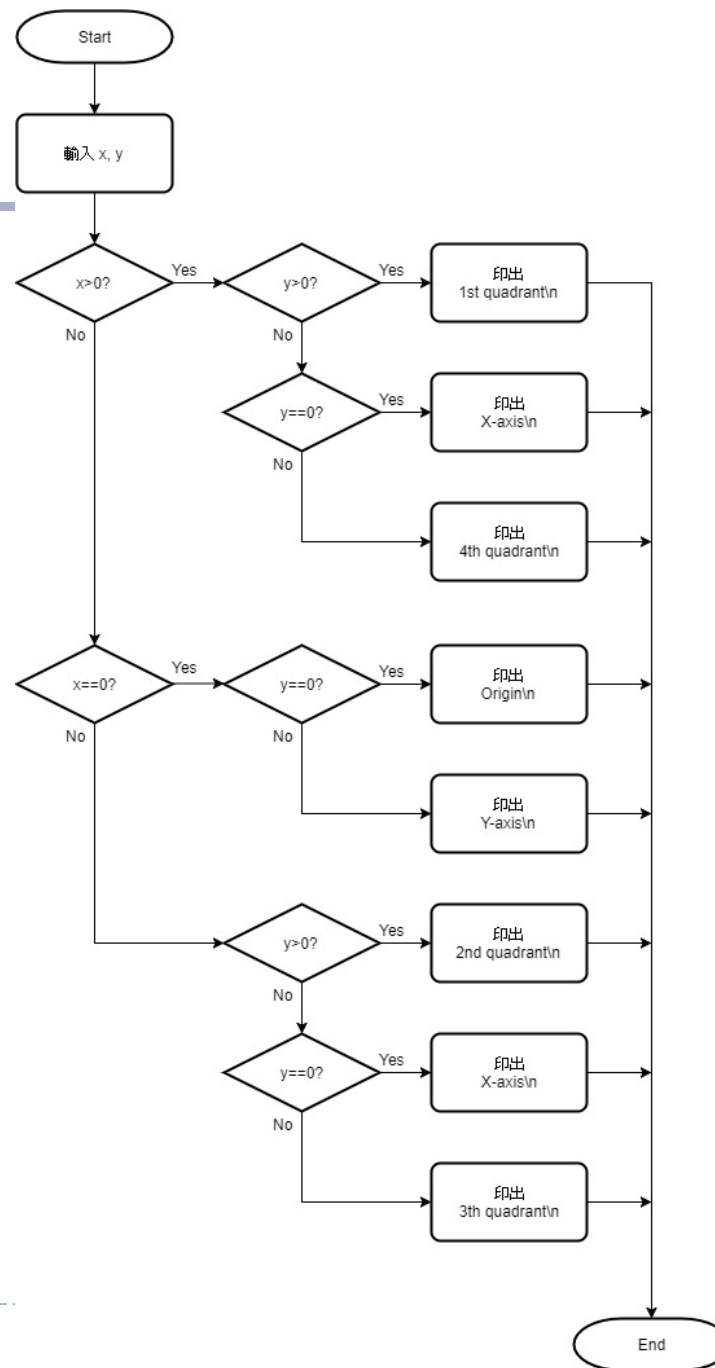
## Ch2. 習題

- ▶ 2-7. 將以下程式的流程圖畫出來。

```
x=int(input("Enter x: "))
y=int(input("Enter y: "))
if x > 0:
    if y > 0:
        print("1st quadrant\n")
    elif y == 0:
        print("X=axis\n")
    else:
        print("4th quadrant\n")
elif x == 0:
    if y == 0:
        print("Origin\n")
    else:
        print("Y=axis\n")
else:
    if y > 0:
        print("2nd quadrant\n")
    elif y == 0:
        print("X-axis\n")
    else:
        print("3rd quadrant\n")
```

## Ch2. 習題

### ► 2-7. 流程圖



休息一下~

---



## Ch3. for迴圈指令

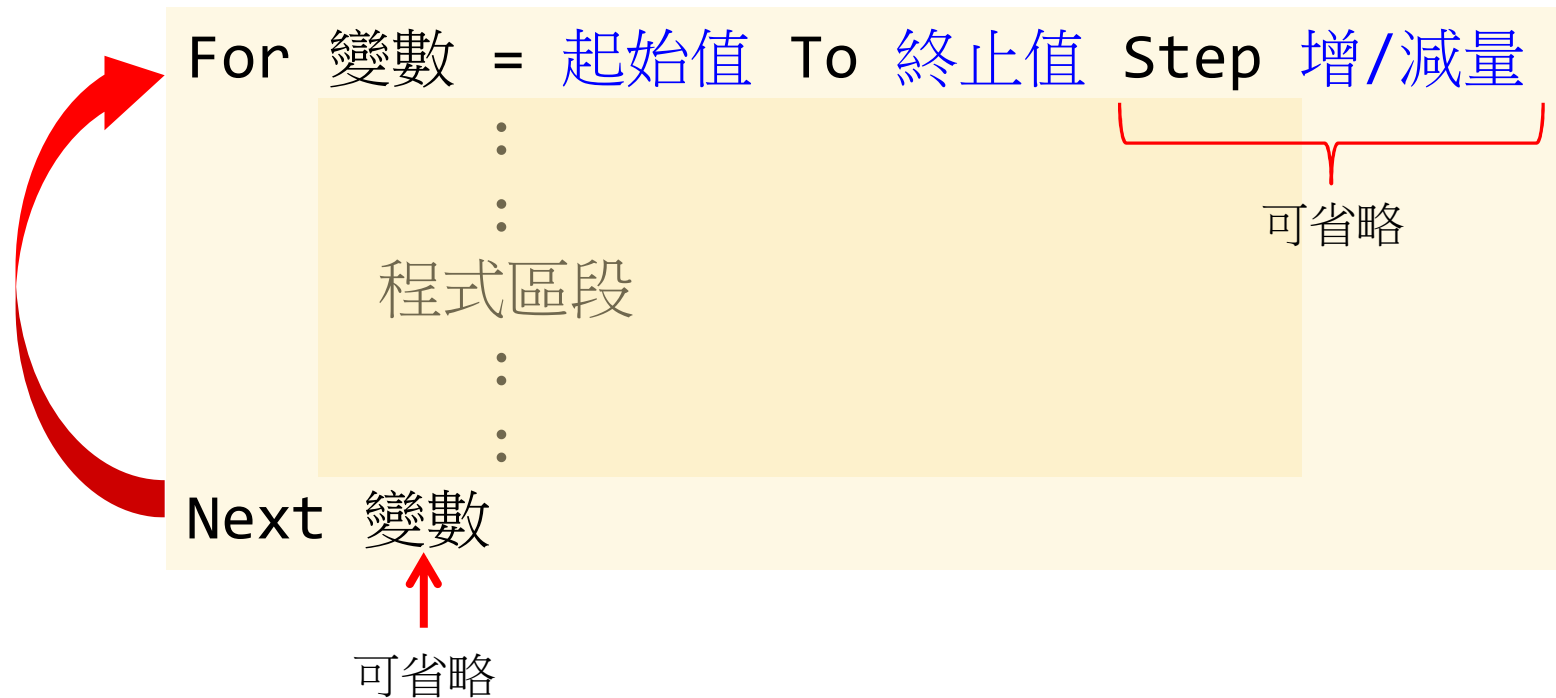
---

- ▶ 如果在程式中遇到要重複同樣動作的指令，就要使用迴圈指令



## Ch3. for迴圈指令

- ▶ 讓程式區段依照給定的起始值及終止值，重複執行多次，語法：



## Ch3. for迴圈指令

- ▶ 讀入五個整數，  
求這五個整數的和。

- ▶ 流程圖：

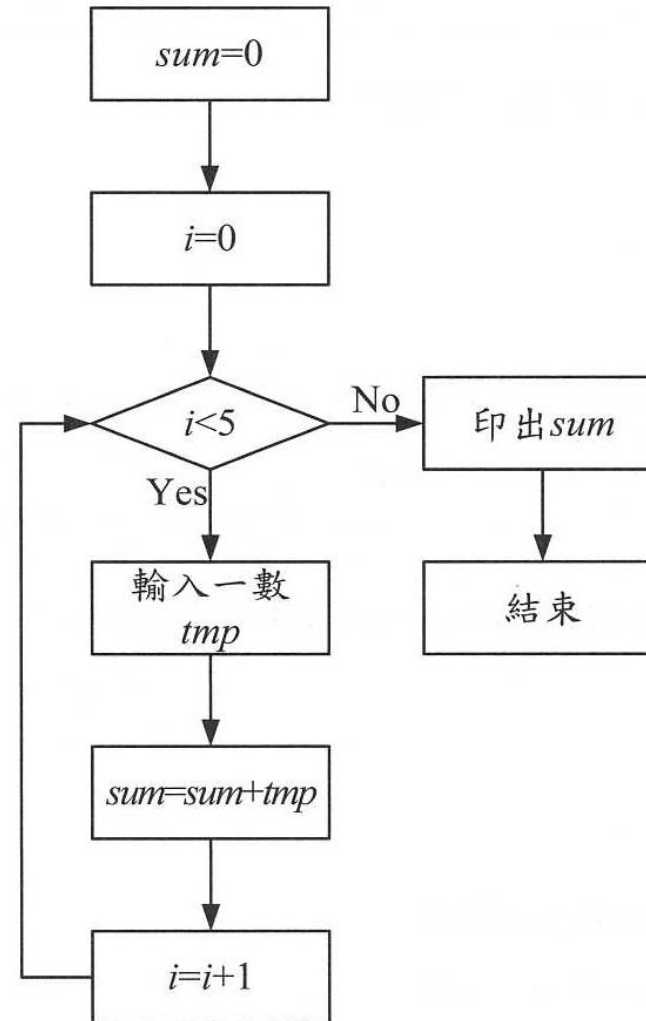


圖 3.1 ▶ 例題 3.1 的流程圖 (五個整數的總和)



## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim sum, tmp As Integer  
    sum = 0  
    For i = 0 To 4  
        Console.WriteLine("Please input the value: ")  
        tmp = Console.ReadLine()  
        sum = sum + tmp  
    Next  
    Console.WriteLine("sum= " & sum)  
    Console.Read()  
End Sub
```

- ▶ i 會從0開始，每次加1，直到i=4為止(共5次)，在這個範圍內for迴圈內的敘述就要重複執行。



## Ch3. for迴圈指令

- ▶ 求N個整數的和，  
N由使用者輸入。

- ▶ 流程圖：

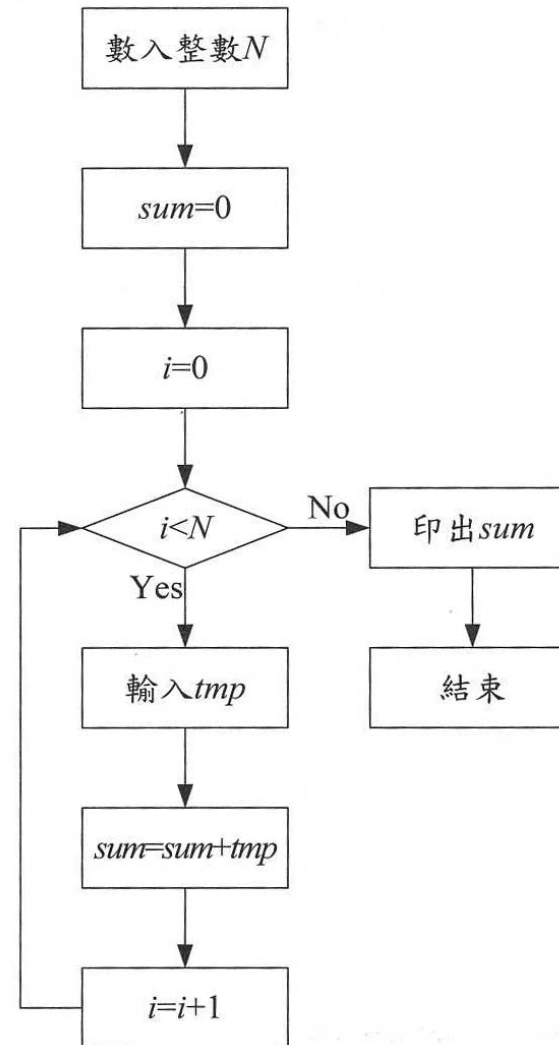


圖 3.2 ▶ 例題 3.2 的流程圖 (N 個數的和)

## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim N, sum, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    sum = 0  
    For i = 1 To N  
        Console.Write("Please input the value: ")  
        tmp = Console.ReadLine()  
        sum = sum + tmp  
    Next  
    Console.WriteLine("sum= " & sum)  
    Console.Read()  
End Sub
```

### ▶ 迴圈重複的次數由輸入的N決定，從1到N



## Ch3. for迴圈指令

---

- ▶ 假設有N個正整數，求其中的最大值。
- ▶ 先假設最大值(MAX)等於0，然後逐一檢查讀入的每一個數字，如果比MAX還大，就令MAX等於這個數字，如果沒有比MAX大，就不做任何事情，最後MAX一定會是這一串數字中的最大值。



## Ch3. for迴圈指令

### ► 流程圖：

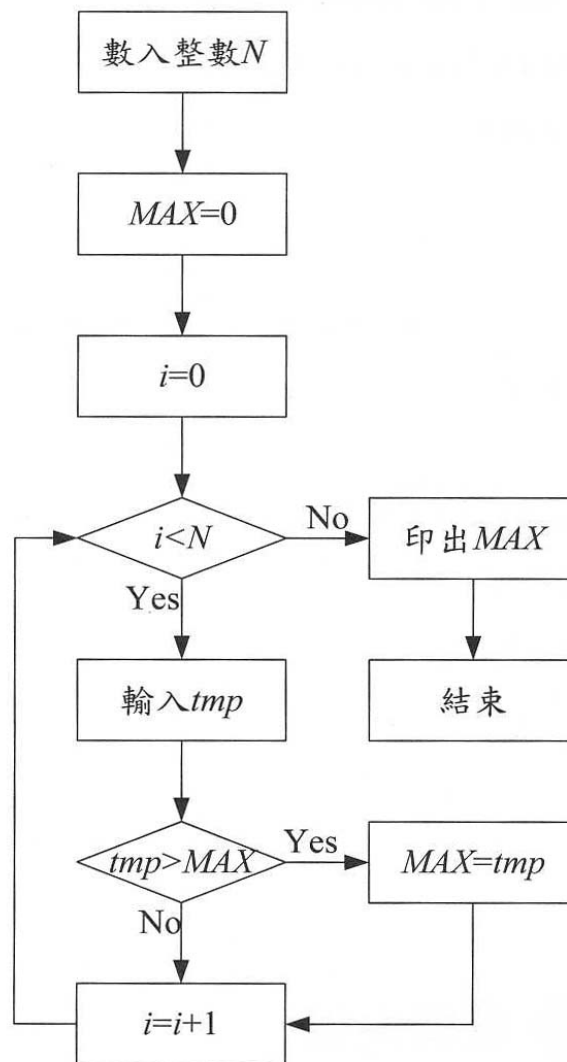


圖 3.3 ► 例題 3.3 的流程圖(求最大值)

## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim N, MAX, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    MAX = 0  
    For i = 1 To N  
        Console.Write("Please input the value: ")  
        tmp = Console.ReadLine()  
        If tmp > MAX Then  
            MAX = tmp  
        End If  
    Next  
    Console.WriteLine("MAX= " & MAX)  
    Console.Read()  
End Sub
```



## Ch3. for迴圈指令

---

- ▶ 求最大奇數值。
- ▶ 先假設最大值(MAX)等於0，然後逐一檢查讀入的每一個數字，如果是奇數且比MAX還大，就令MAX等於這個數字，如果不是，就不做任何事情，最後MAX一定會是這一串數字中的奇數最大值。
- ▶ 可以用%(取餘數)運算，例如將n除以2之後若餘0，為偶數，若餘1則為奇數。



## Ch3. for迴圈指令

### ► 流程圖：

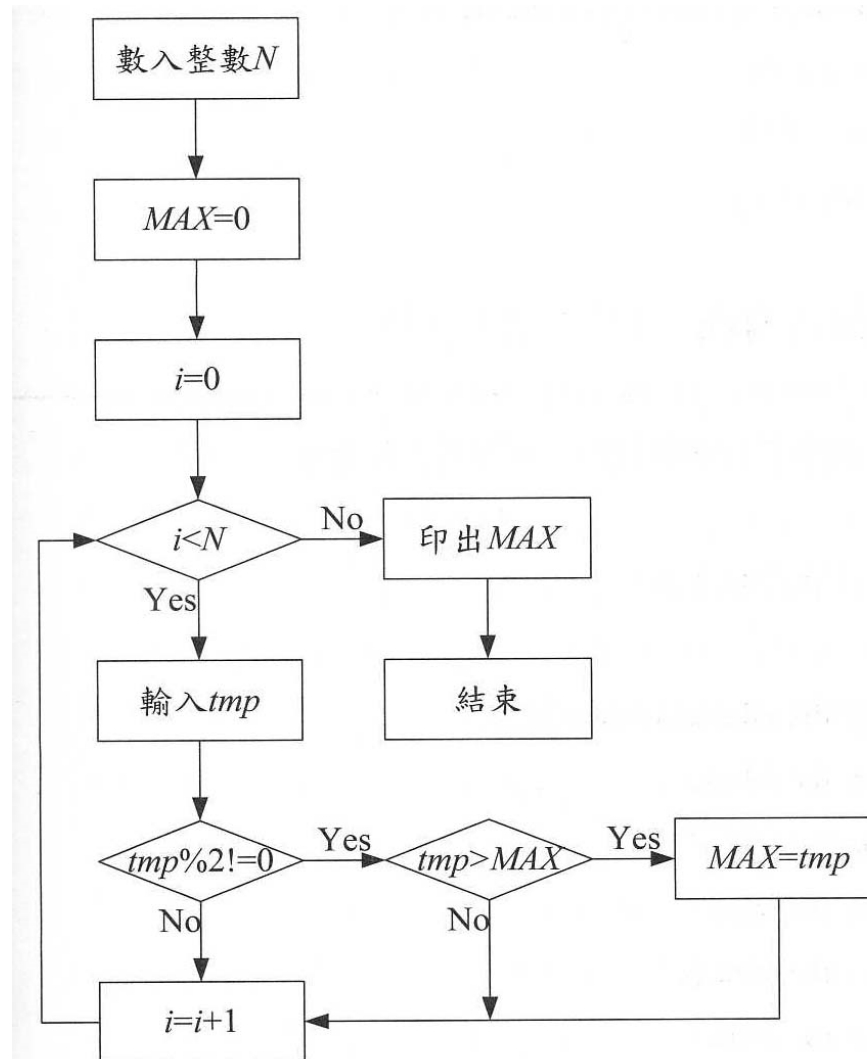


圖 3.4 ► 例題 3.4 的流程圖: 求最大奇數值



## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim N, MAX, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    MAX = 0  
    For i = 1 To N  
        Console.Write("Please input the value: ")  
        tmp = Console.ReadLine()  
        If tmp Mod 2 <> 0 Then '除2的餘數不為0的就是奇數  
            If tmp > MAX Then  
                MAX = tmp  
            End If  
        End If  
    Next  
    Console.WriteLine("MAX= " & MAX)  
    Console.Read()  
End Sub
```

## Ch3. for迴圈指令

- ▶ 求N階層，即求 $1 \times 2 \times 3 \times \dots \times N$ 。
- ▶ 請求輸入N，讓for迴圈自2依序乘到N，所以要寫成

for i = 2 To N

- ▶ 流程圖：

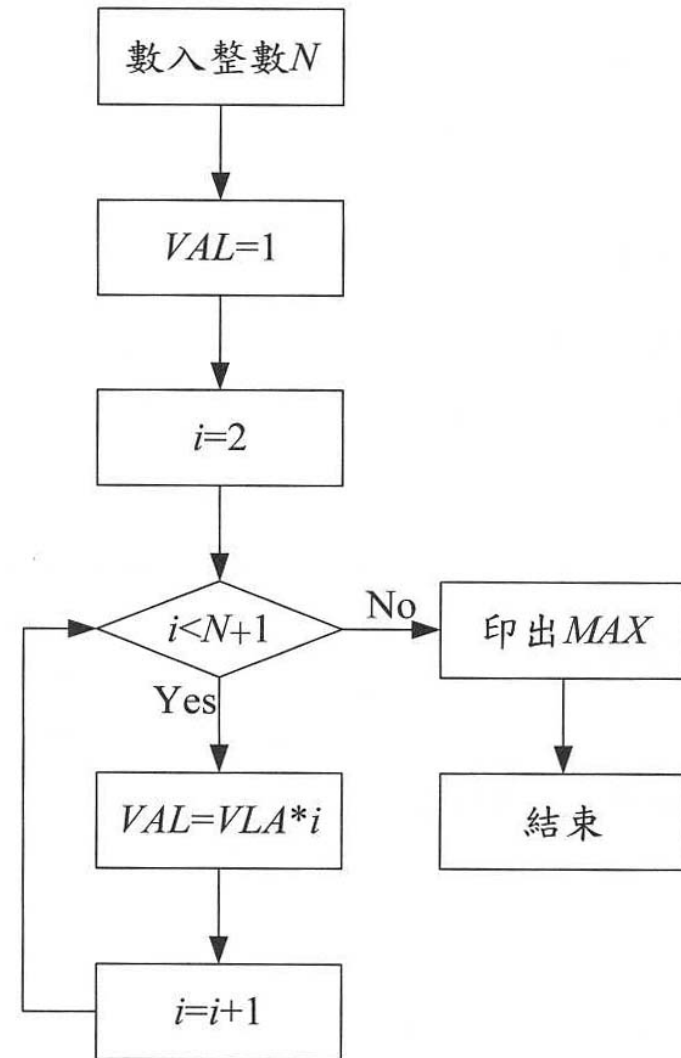


圖 3.5 ▶ 例題 3.5 的流程圖 (求 N 的階層)

## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim N, VAL As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    VAL = 1  
    For i = 2 To N  
        VAL = VAL * i  
    Next  
    Console.WriteLine("VAL= " & VAL)  
    Console.Read()  
End Sub
```

### ▶ VAL = VAL \* i 也可寫成 VAL \*= i



## Ch3. for迴圈指令

- ▶ 輸出一個九九乘法表。
- ▶ 利用兩層迴圈，又稱巢狀迴圈。
- ▶ 迴圈的層數沒有限制，但越多效率越差。
- ▶ 流程圖：

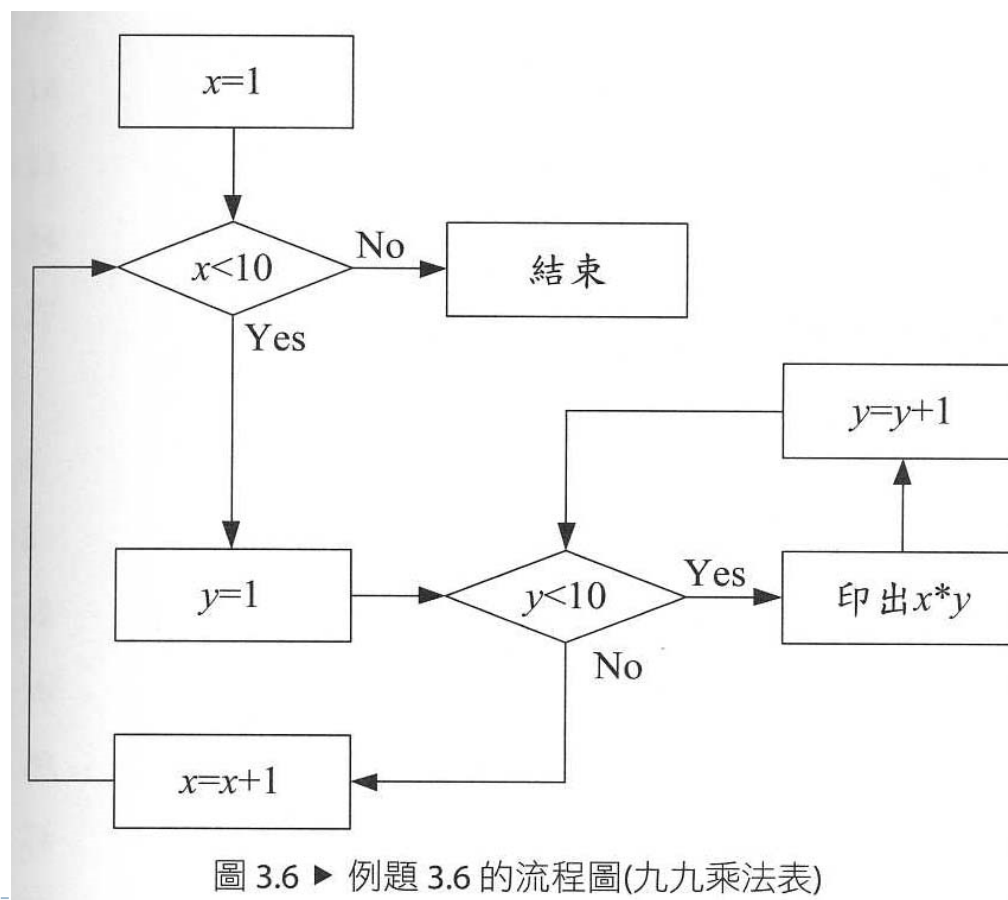


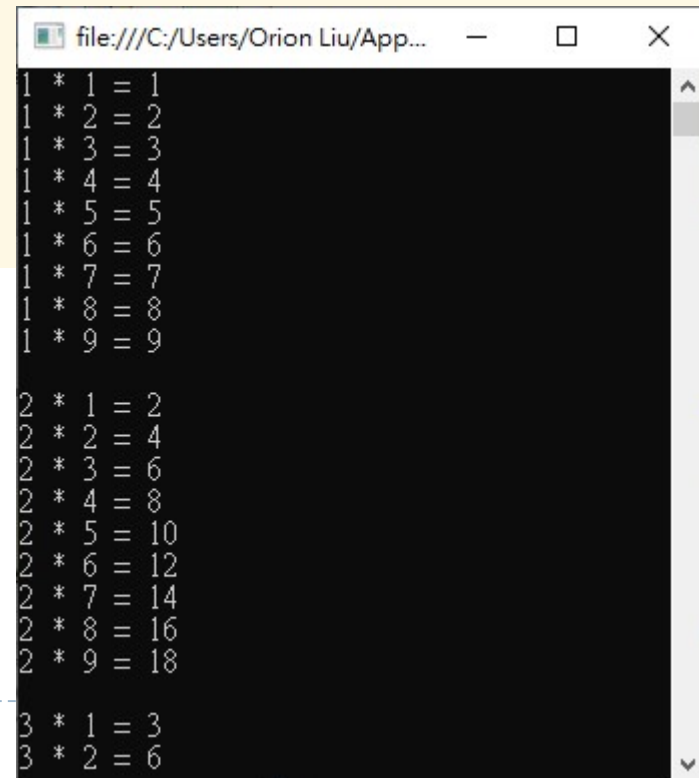
圖 3.6 ▶ 例題 3.6 的流程圖(九九乘法表)

## Ch3. for迴圈指令

### ▶ 參考程式：

```
Sub Main()  
    For i = 1 To 9  
        For j = 1 To 9  
            Console.WriteLine(i & " * " & j & " = " & i * j)  
        Next  
        Console.WriteLine()  
    Next  
    Console.Read()  
End Sub
```

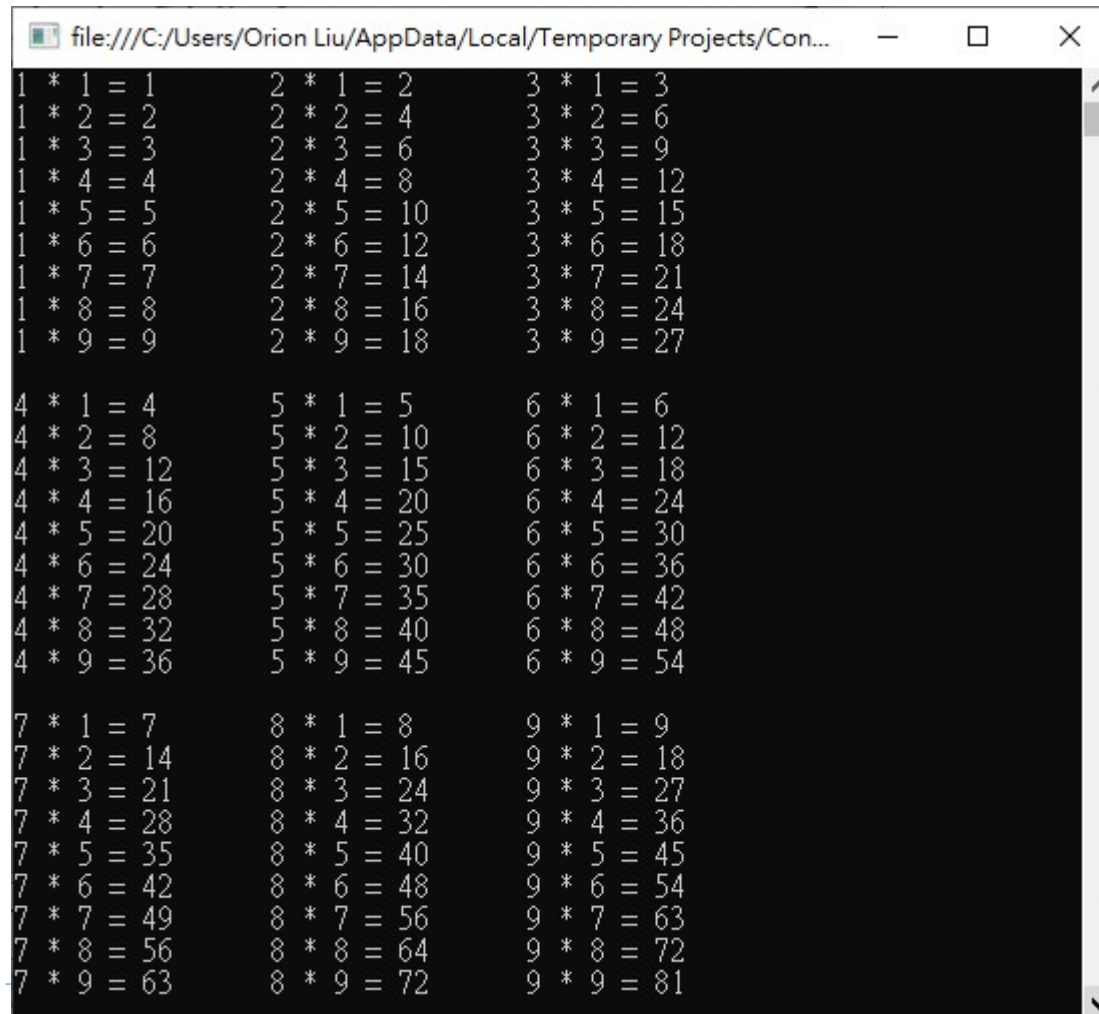
### ▶ 執行結果：



```
file:///C:/Users/Orion Liu/App...  
1 * 1 = 1  
1 * 2 = 2  
1 * 3 = 3  
1 * 4 = 4  
1 * 5 = 5  
1 * 6 = 6  
1 * 7 = 7  
1 * 8 = 8  
1 * 9 = 9  
  
2 * 1 = 2  
2 * 2 = 4  
2 * 3 = 6  
2 * 4 = 8  
2 * 5 = 10  
2 * 6 = 12  
2 * 7 = 14  
2 * 8 = 16  
2 * 9 = 18  
  
3 * 1 = 3  
3 * 2 = 6
```

## Ch3. for迴圈指令

- ▶ 請將程式改成如下輸出：



```
file:///C:/Users/Orion Liu/AppData/Local/Temporary Projects/Con...
1 * 1 = 1    2 * 1 = 2    3 * 1 = 3
1 * 2 = 2    2 * 2 = 4    3 * 2 = 6
1 * 3 = 3    2 * 3 = 6    3 * 3 = 9
1 * 4 = 4    2 * 4 = 8    3 * 4 = 12
1 * 5 = 5    2 * 5 = 10   3 * 5 = 15
1 * 6 = 6    2 * 6 = 12    3 * 6 = 18
1 * 7 = 7    2 * 7 = 14    3 * 7 = 21
1 * 8 = 8    2 * 8 = 16    3 * 8 = 24
1 * 9 = 9    2 * 9 = 18    3 * 9 = 27

4 * 1 = 4    5 * 1 = 5    6 * 1 = 6
4 * 2 = 8    5 * 2 = 10   6 * 2 = 12
4 * 3 = 12   5 * 3 = 15   6 * 3 = 18
4 * 4 = 16   5 * 4 = 20   6 * 4 = 24
4 * 5 = 20   5 * 5 = 25   6 * 5 = 30
4 * 6 = 24   5 * 6 = 30   6 * 6 = 36
4 * 7 = 28   5 * 7 = 35   6 * 7 = 42
4 * 8 = 32   5 * 8 = 40   6 * 8 = 48
4 * 9 = 36   5 * 9 = 45   6 * 9 = 54

7 * 1 = 7    8 * 1 = 8    9 * 1 = 9
7 * 2 = 14   8 * 2 = 16   9 * 2 = 18
7 * 3 = 21   8 * 3 = 24   9 * 3 = 27
7 * 4 = 28   8 * 4 = 32   9 * 4 = 36
7 * 5 = 35   8 * 5 = 40   9 * 5 = 45
7 * 6 = 42   8 * 6 = 48   9 * 6 = 54
7 * 7 = 49   8 * 7 = 56   9 * 7 = 63
7 * 8 = 56   8 * 8 = 64   9 * 8 = 72
7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
```

## Ch3. for迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    For i = 1 To 9 Step 3  
        For j = 1 To 9  
            Console.Write(i & " * " & j & " = " & i * j & vbCrLf)  
            Console.Write(i + 1 & " * " & j & " = " & (i + 1) * j & vbCrLf)  
            Console.WriteLine(i + 2 & " * " & j & " = " & (i + 2) * j)  
        Next  
        Console.WriteLine()  
    Next  
    Console.Read()  
End Sub
```


- ▶ vbCrLf是Tab，跳格的意思。
- ▶ 常用的還有vbCrLf，是換行的意思。
- ▶ Console.WriteLine()函數結尾會加上換行，Console.Write()則不會。



## For ... Next敘述

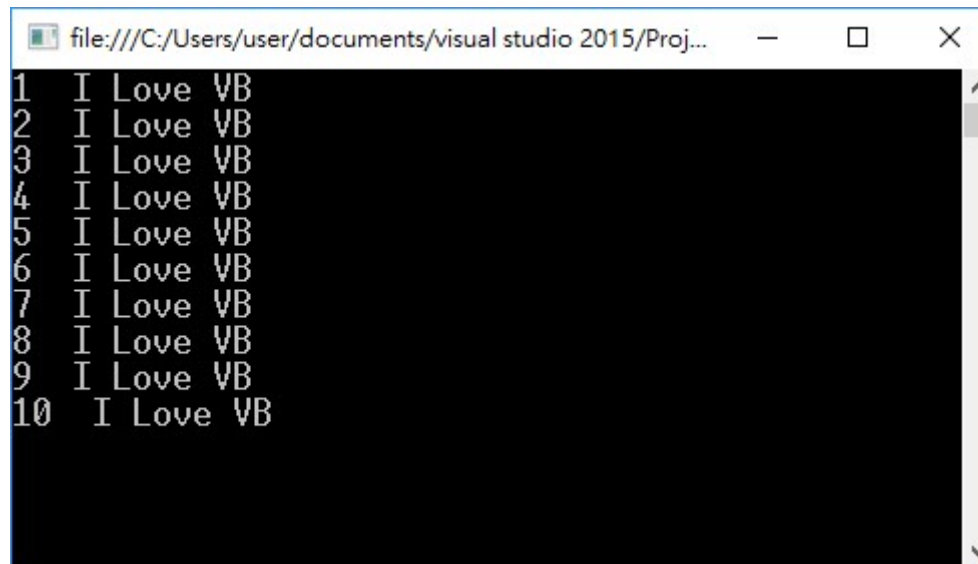
---

- ▶ 範例練習，印出10次“I Love VB”：



```
For I = 1 To 10 Step 1
    Console.WriteLine(I & “I Love VB”)
Next I
```

- ▶ 執行結果：




```
1 I Love VB
2 I Love VB
3 I Love VB
4 I Love VB
5 I Love VB
6 I Love VB
7 I Love VB
8 I Love VB
9 I Love VB
10 I Love VB
```



# For ... Next敘述

---

變數 = 起始值 To 終止值 Step 增/減量



```
For I = 1 To 10 Step 1
    Console.WriteLine(I & " I Love VB")
Next I
```

- ▶ I = 1時，印一行 I Love VB
- ▶ I = 2時，印一行 I Love VB
- ▶ :
- ▶ I = 10時，印一行 I Love VB
- ▶ 一共印出10行

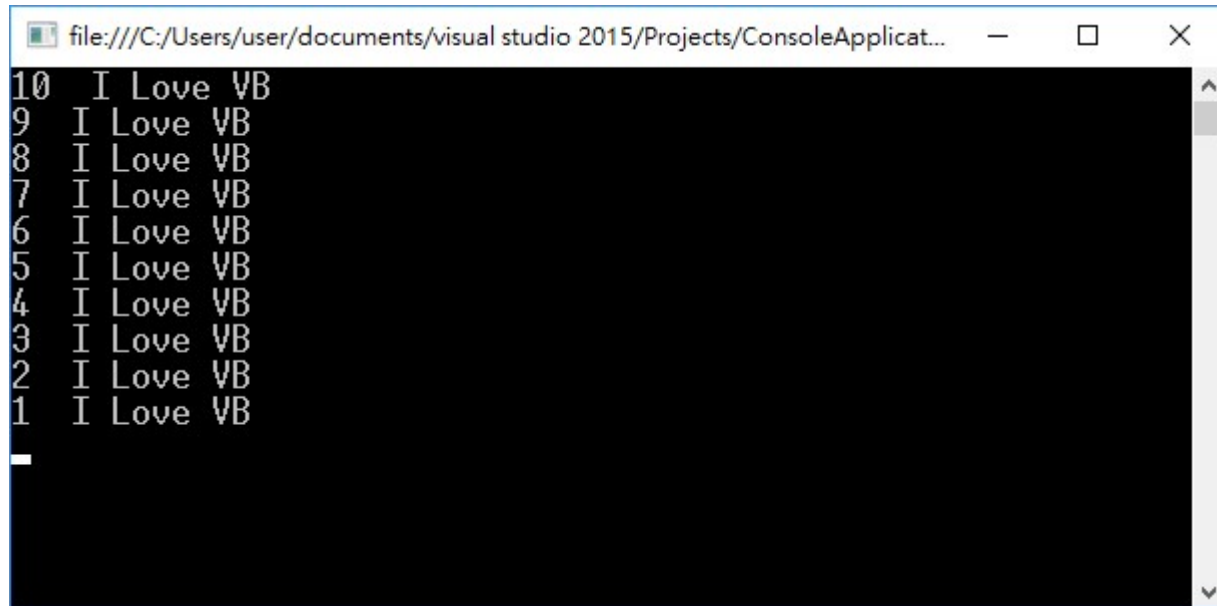
## For ... Next敘述

---

- ▶ 也可以用倒數的：

```
For I = 10 To 1 Step -1  
    Console.WriteLine(I & " I Love VB")  
Next I
```

- ▶ 執行結果：



A screenshot of a Windows console application window. The title bar shows the file path: file:///C:/Users/user/documents/visual studio 2015/Projects/ConsoleApplicat... The console output displays ten lines, each containing a number from 10 down to 1, followed by the text " I Love VB". The numbers are aligned to the left, and the text is indented. A cursor is visible at the bottom of the console window.

```
10 I Love VB  
9 I Love VB  
8 I Love VB  
7 I Love VB  
6 I Love VB  
5 I Love VB  
4 I Love VB  
3 I Love VB  
2 I Love VB  
1 I Love VB
```

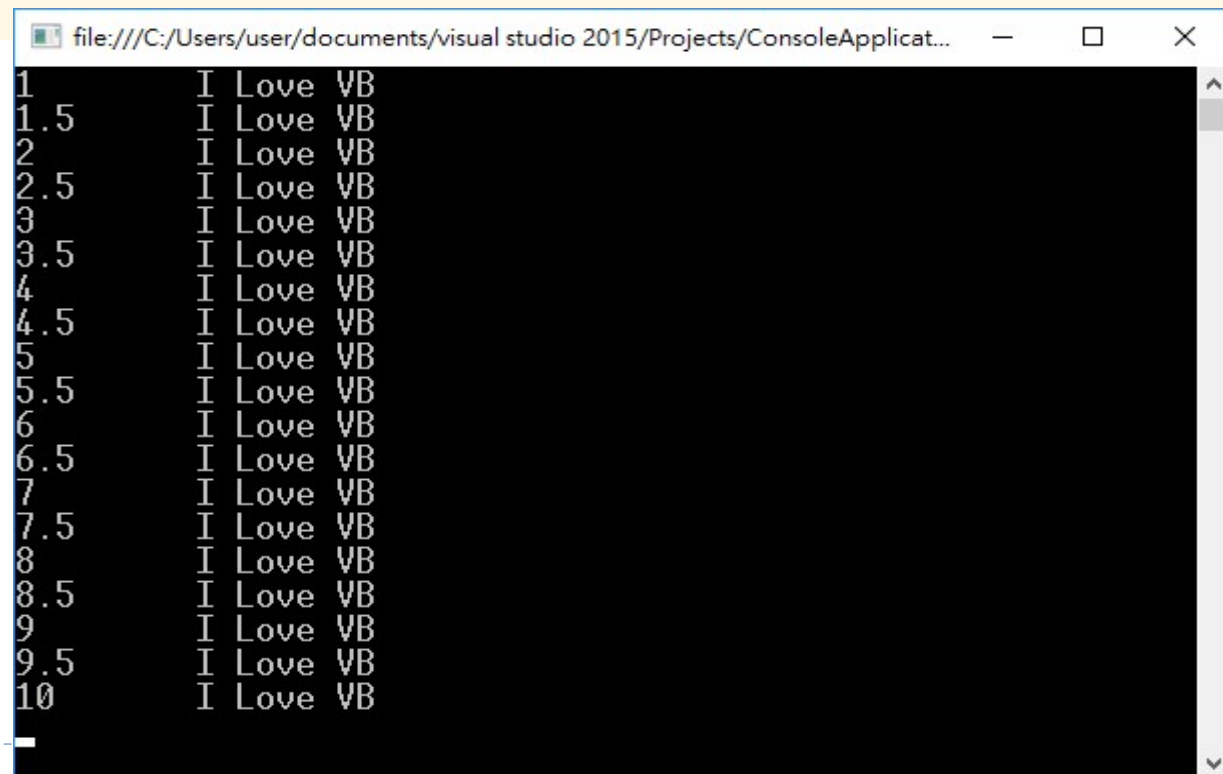
## For ... Next敘述

- ▶ 增/減量也可以不是整數：

插入一個跳格(Tab)

```
For I = 1 To 10 Step 0.5
    Console.WriteLine(I & vbTab & " I Love VB")
Next I
```

- ▶ 執行結果：  
(20次)



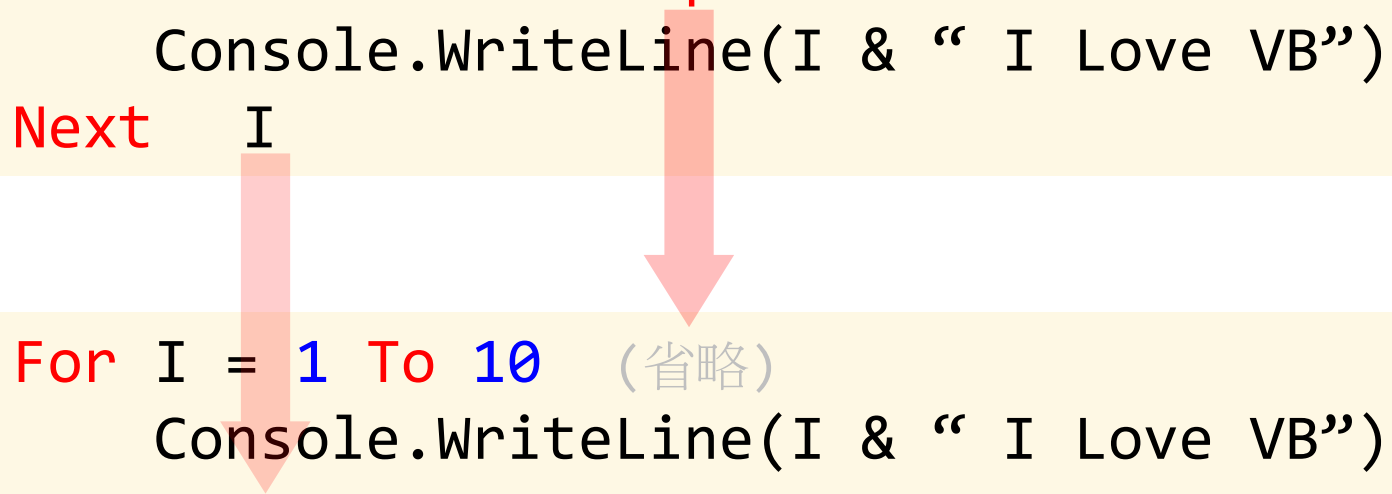
```
1 I Love VB
1.5 I Love VB
2 I Love VB
2.5 I Love VB
3 I Love VB
3.5 I Love VB
4 I Love VB
4.5 I Love VB
5 I Love VB
5.5 I Love VB
6 I Love VB
6.5 I Love VB
7 I Love VB
7.5 I Love VB
8 I Love VB
8.5 I Love VB
9 I Love VB
9.5 I Love VB
10 I Love VB
```

## For ... Next敘述

---

- ▶ 增/減量是+1時可省略，Next後面的變數可省略：

```
For I = 1 To 10 Step 1  
    Console.WriteLine(I & " I Love VB")  
Next I
```



```
For I = 1 To 10 (省略)  
    Console.WriteLine(I & " I Love VB")  
Next (省略)
```


在多重迴圈中若需要辨識，  
還是可以寫出來

## For ... Next敘述

---


- ▶ 錯誤的用法：(一次都不會執行)

```
For I = 10 To 1 Step 1  
    Console.WriteLine(I & " I Love VB")  
Next I
```



- ▶ 或

```
For I = 1 To 10 Step -1  
    Console.WriteLine(I & " I Love VB")  
Next I
```

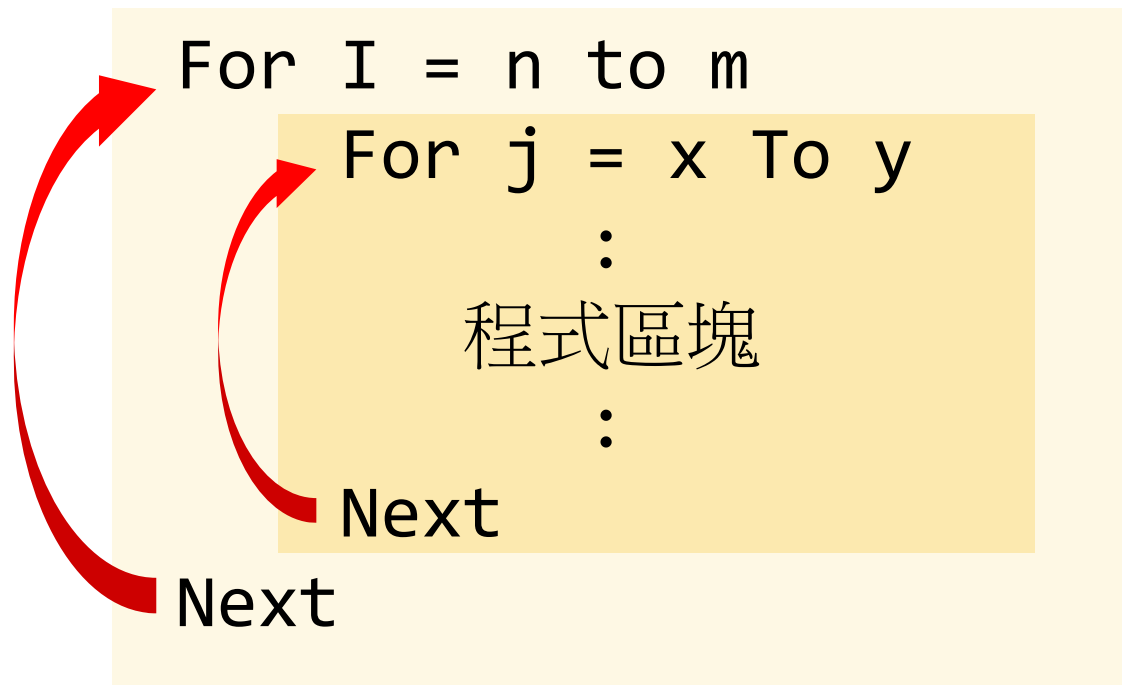


- ▶ 原因：增/減量無法讓初始值趨向終止值

## For 巢狀迴圈

---

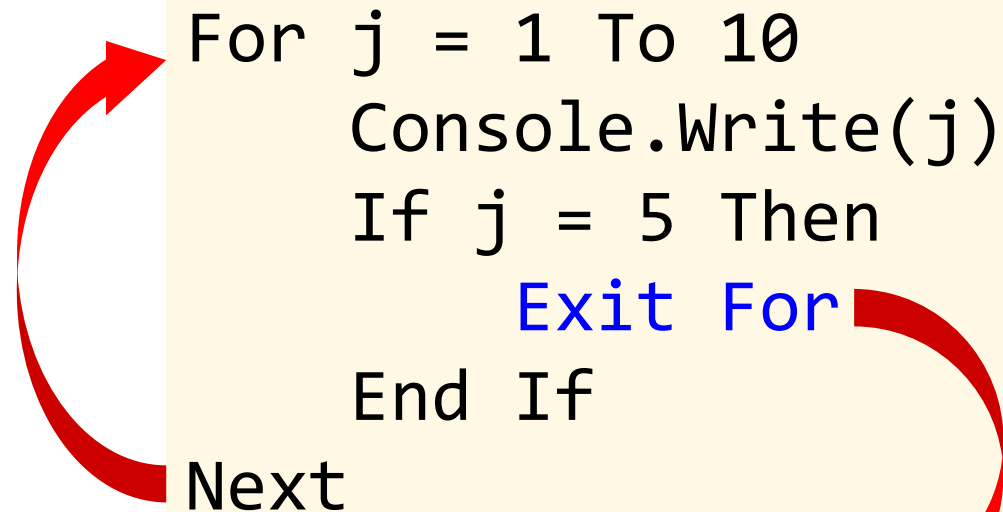
- ▶ 就是迴圈裏面還有迴圈



- ▶ 注意：套得越多層，效率越差

## Exit For敘述

- ▶ 此敘述可以強迫跳出所在的迴圈，不執行剩下的敘述。
- ▶ 範例：



```
For j = 1 To 10
  Console.Write(j)
  If j = 5 Then
    Exit For
  End If
Next
```

執行迴圈外的下一道敘述

跳出迴圈，

## Exit For敘述

---

- ▶ Exit For只能跳出所在的迴圈，不會一次全跳出來。

```
For i = 1 To 10
    For j = 1 To 10
        Console.Write(j)
        If j = 5 Then
            Exit For
        End If
    Next
    Console.WriteLine()
Next
```



## Ch3. 習題

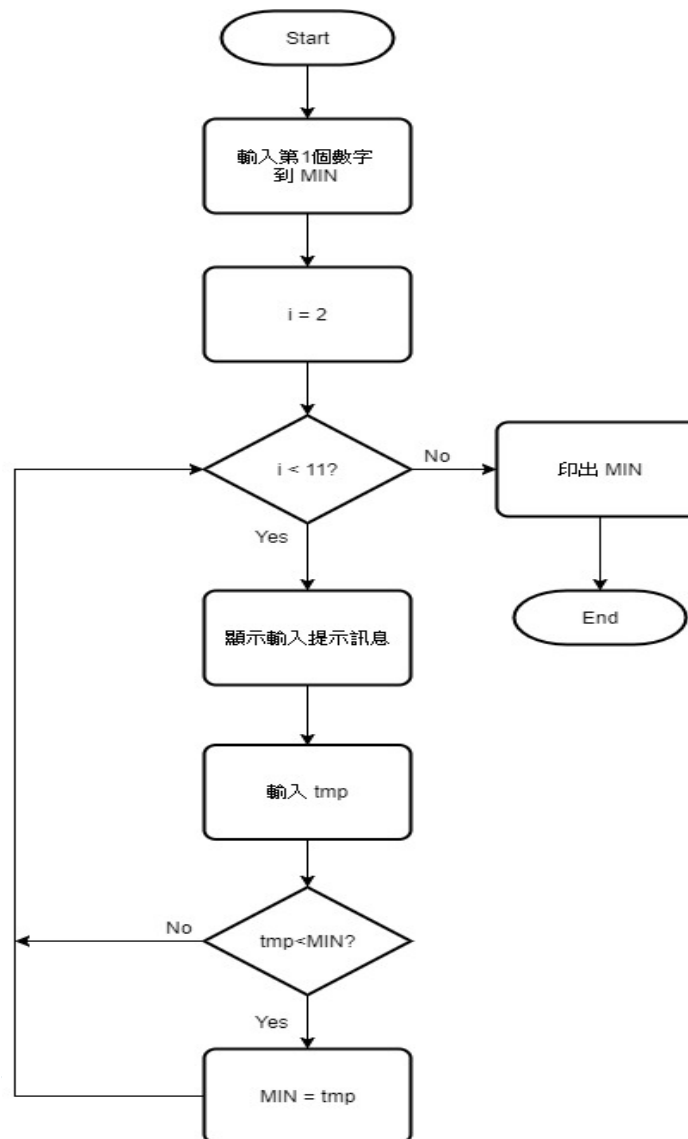
---

- ▶ 3-1. 寫一程式，輸入10個整數，求其最小值。
- ▶ 3-2. 寫一程式，輸入N個整數，求其最小值。
- ▶ 3-3. 寫一程式，輸入10個整數，列出其中所有大於12的數字。
- ▶ 3-4. 寫一程式，輸入10個整數，列出其中所有大於12的數字的總和。



## Ch3. 習題

### ► 3-1 流程圖：



## Ch3. 習題

---

### ▶ 3-1參考程式：

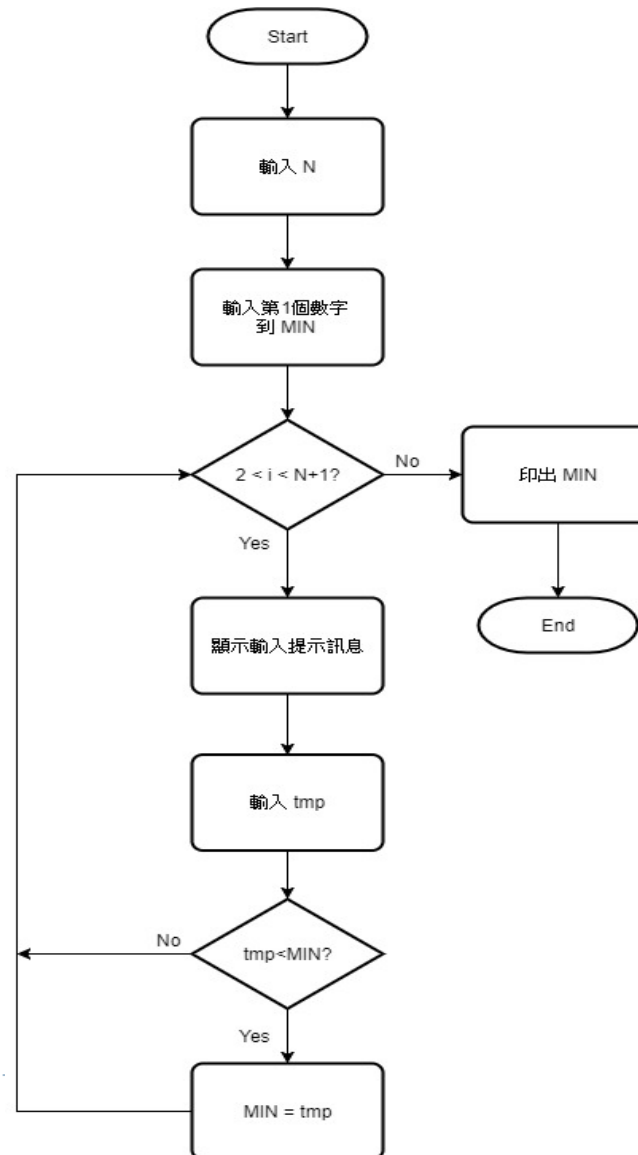
```
Sub Main()  
    Dim MIN, tmp As Integer  
    Console.Write("Please input the value of 1 : ")  
    MIN = Console.ReadLine()  
    For i = 2 To 10  
        Console.Write("Please input the value of " & i & ": ")  
        tmp = Console.ReadLine()  
        If tmp < MIN Then  
            MIN = tmp  
        End If  
    Next  
    Console.WriteLine("MIN = " & MIN)  
    Console.Read()  
End Sub
```

- ▶ 先將第一個輸入的值當成最小的，再依序跟後續輸入的做比較。



## Ch3. 習題

### ► 3-2 流程圖：



## Ch3. 習題

---

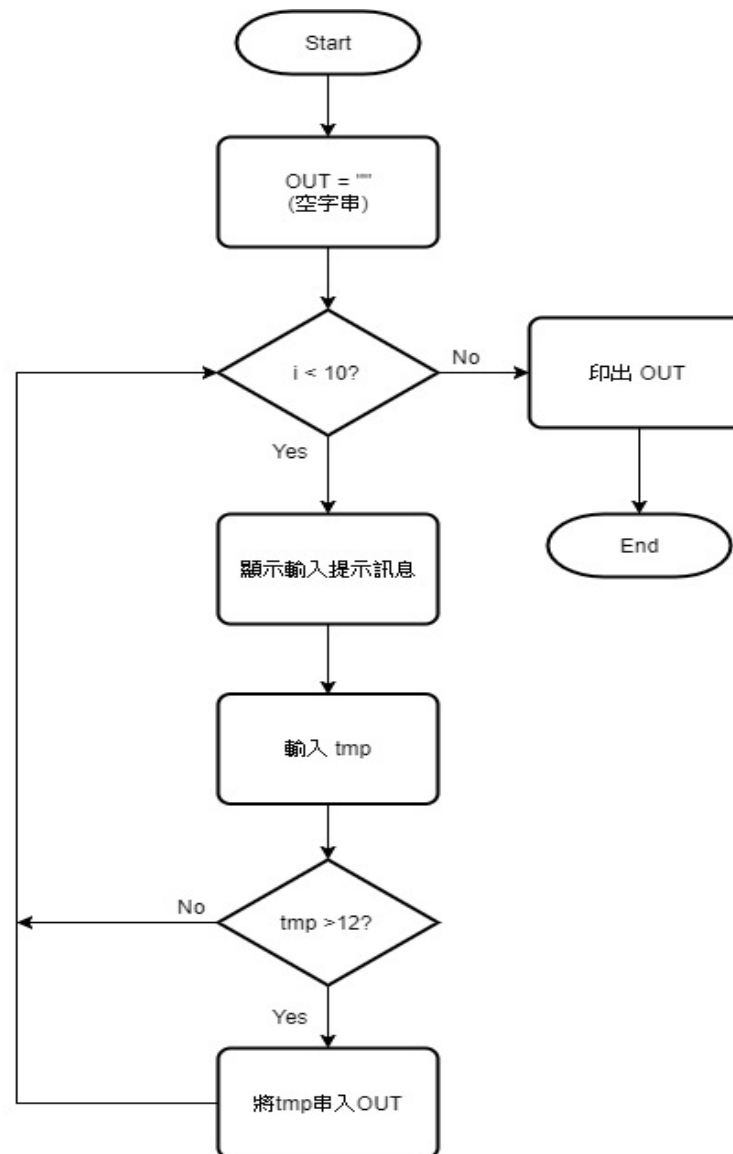
### ▶ 3-2參考程式：

```
Sub Main()  
    Dim N, MIN, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    Console.Write("Please input the value of 1 : ")  
    MIN = Console.ReadLine()  
    For i = 2 To N  
        Console.Write("Please input the value of " & i & " : ")  
        tmp = Console.ReadLine()  
        If tmp < MIN Then  
            MIN = tmp  
        End If  
    Next  
    Console.WriteLine("MIN = " & MIN)  
    Console.Read()  
End Sub
```



## Ch3. 習題

### ► 3-3 流程圖：



## Ch3. 習題

---

### ▶ 3-3參考程式：

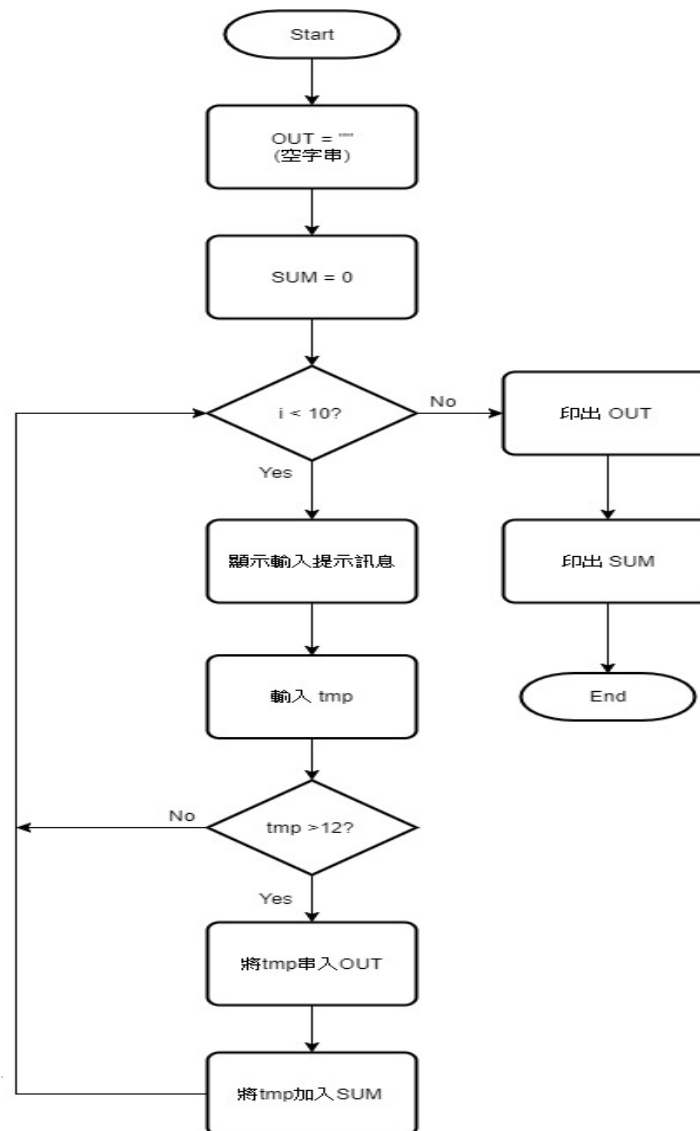
```
Sub Main()  
    Dim tmp As Integer  
    Dim OUT As String  
    OUT = "" '先清空字串變數  
    For i = 1 To 10  
        Console.Write("Please input number " & i & " : ")  
        tmp = Console.ReadLine()  
        If tmp > 12 Then  
            OUT = OUT & tmp & " "  
        End If  
    Next  
    Console.WriteLine("Number bigger then 12: " & OUT)  
    Console.Read()  
End Sub
```

- ▶ 將符合大於12的數字轉成文字串入OUT，最後將OUT印出來。
- 



## Ch3. 習題

### ► 3-4 流程圖：





## Ch3. 習題

---

### ▶ 3-4參考程式：

```
Sub Main()  
    Dim tmp, SUM As Integer  
    Dim OUT As String  
    OUT = ""      '先清空字串變數  
    SUM = 0       'SUM初始值為0  
    For i = 1 To 10  
        Console.Write("Please input number " & i & " : ")  
        tmp = Console.ReadLine()  
        If tmp > 12 Then  
            OUT = OUT & tmp & " "  
            SUM += tmp  
        End If  
    Next  
    Console.WriteLine("Number bigger then 12: " & OUT)  
    Console.WriteLine("Sum = " & SUM)  
    Console.Read()  
End Sub
```

## Ch3. 習題

---

- ▶ 3-5. 寫一程式，輸入N個數字，求其所有奇數中的最大值。例如輸入11,12,3,24,15，答案是15。

- ▶ 3-6. 寫一程式，輸入N個數字，求其所有正數之平方的加總。例如輸入1,-2,3,-4,5五個數字，得到

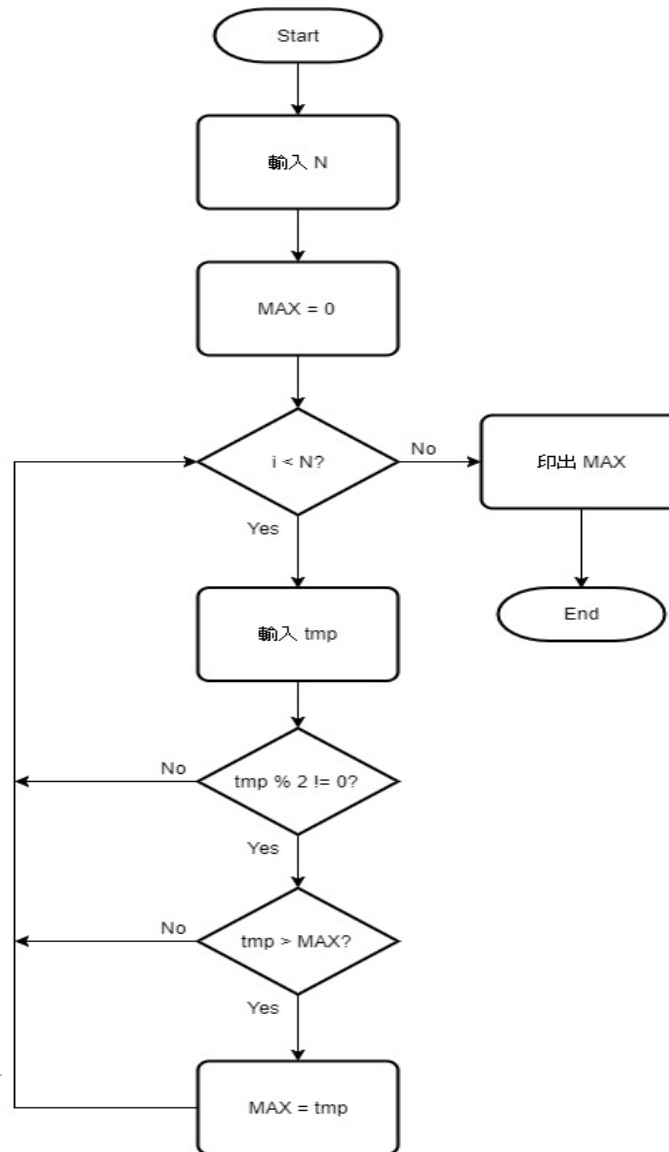
$$1^2 + (-2)^2 + 3^2 + (-4)^2 + 5^2 = 1 + 9 + 25 = 35$$

- ▶ 3-7. 寫一程式，輸入N個數字，其中有些是負數，將這些負數轉換成正數，例如-7會被轉換成7。



## Ch3. 習題

### ► 3-5. 流程圖：



## Ch3. 習題

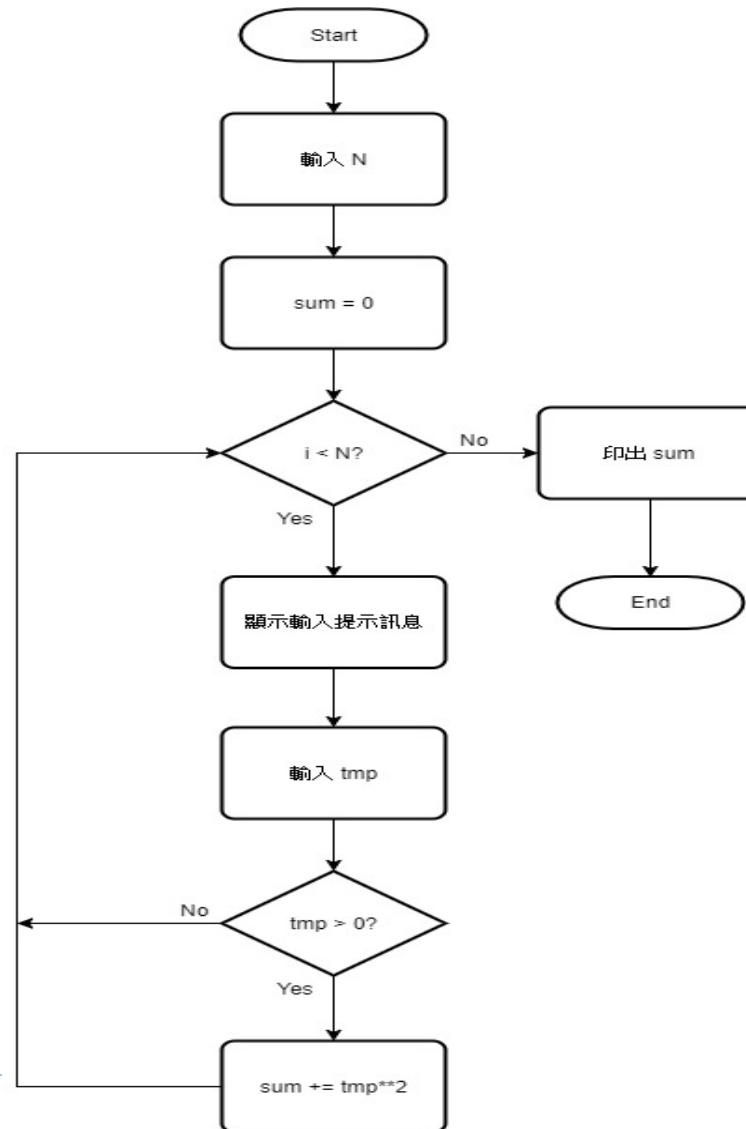
---

### ▶ 3-5. 參考程式：

```
Sub Main()  
    Dim N, MAX, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    MAX = 0  
    For i = 1 To N  
        Console.Write("Please input the value: ")  
        tmp = Console.ReadLine()  
        If tmp Mod 2 <> 0 Then '是否為奇數  
            If tmp > MAX Then '是否大於MAX  
                MAX = tmp '是則紀錄tmp為目前最大值  
            End If  
        End If  
    Next  
    Console.WriteLine("MAX = " & MAX)  
    Console.Read()  
End Sub
```

## Ch3. 習題

### ► 3-6. 流程圖：



## Ch3. 習題

---

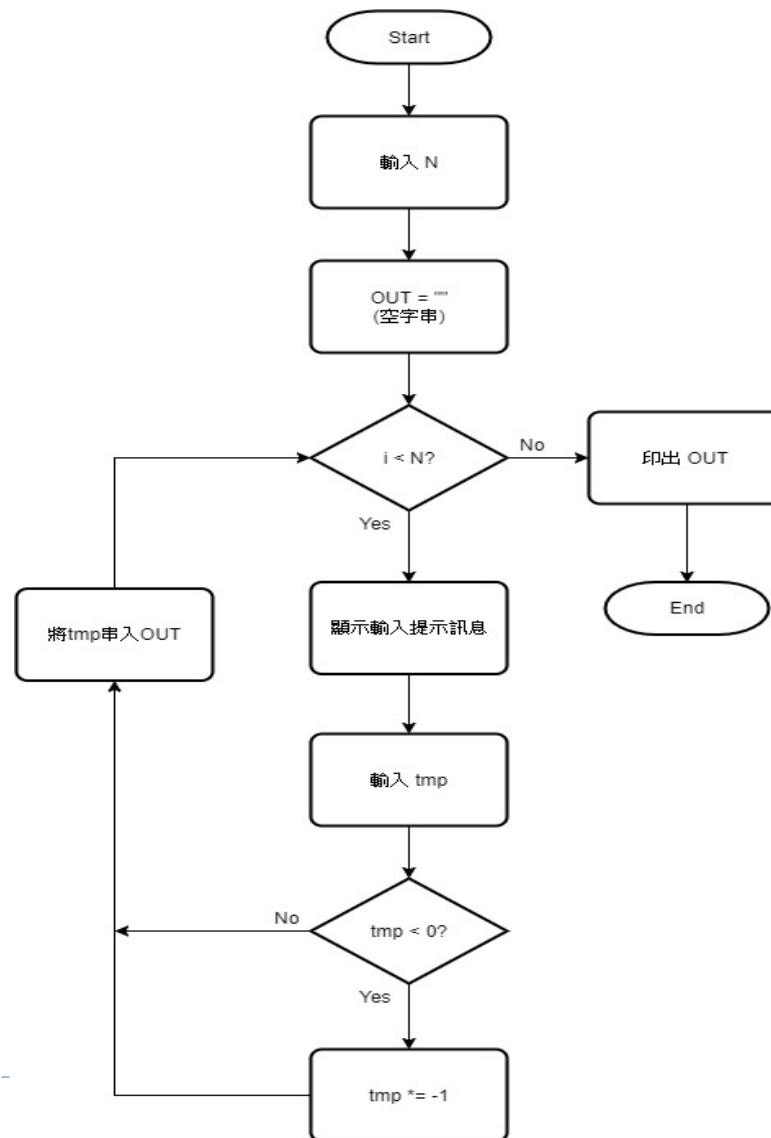
### ▶ 3-6. 參考程式：

```
Sub Main()  
    Dim N, sum, tmp As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    sum = 0  
    For i = 1 To N  
        Console.Write("Please input number: " & i & ": ")  
        tmp = Console.ReadLine()  
        If tmp > 0 Then  
            sum += tmp ^ 2  
        End If  
    Next  
    Console.WriteLine("Sum = " & sum)  
    Console.Read()  
End Sub
```



## Ch3. 習題

### ► 3-7. 流程圖：



## Ch3. 習題

---

### ▶ 3-7. 參考程式：

```
Sub Main()  
    Dim N, tmp As Integer  
    Dim OUT As String  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    OUT = ""  
    For i = 1 To N  
        Console.Write("Please input number: " & i & ": ")  
        tmp = Console.ReadLine()  
        If tmp < 0 Then  
            tmp *= -1  
        End If  
        OUT = OUT & tmp & " "  
    Next  
    Console.WriteLine("Result = " & OUT)  
    Console.Read()  
End Sub
```



休息一下~

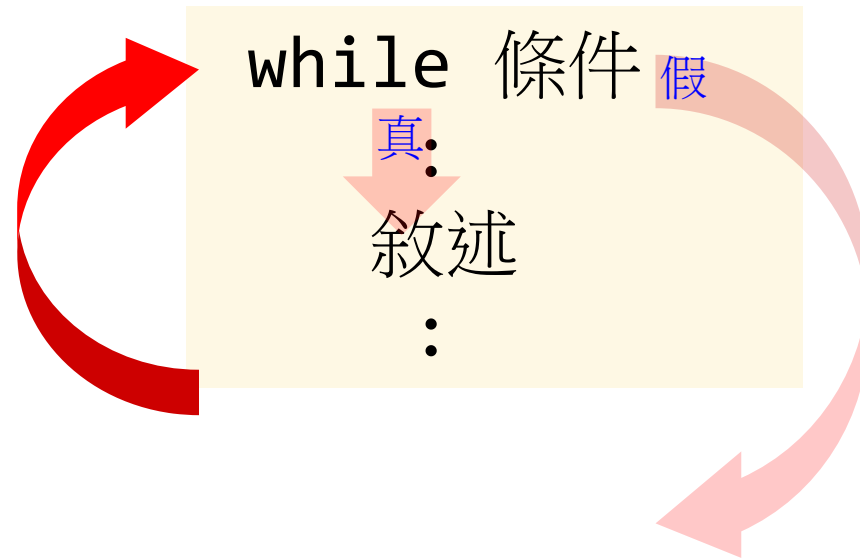
---



## Ch4. while迴圈指令

---

- ▶ 通常for圈都用在確定要重複執行多少次的情況，若重複的次數不定，我們可以用while迴圈。
- ▶ while以”是”、“否”來做為終止條件。
- ▶ 當條件為真時執行迴圈，為假時結束迴圈



## Ch4. while迴圈指令

- ▶ 讀入一連串的數字，但事先不知道數字的數目。
- ▶ 設定一個特殊的數字，一旦讀到這個數字就終止迴圈。
- ▶ 假設輸入的都是正數，一旦讀到負數就終止。
- ▶ 流程圖：

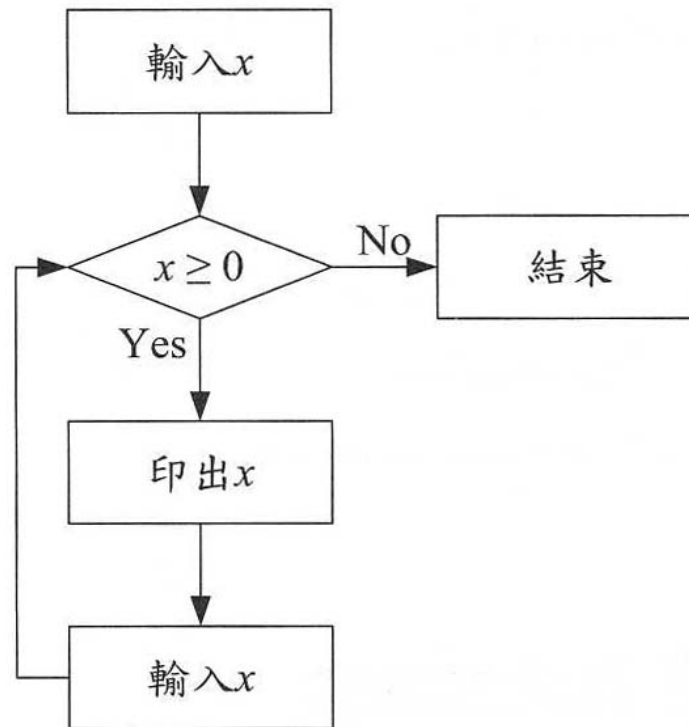


圖 4.1 ▶ 例題 4.1 的流程圖

## Ch4. while迴圈指令

---

▶ 參考程式：

```
Sub Main()  
    Dim x As Integer  
    Console.Write("Please input the value of variable: ")  
    x = Console.ReadLine()  
    While x >= 0  
        Console.WriteLine("x = " & x)  
        Console.Write("Please input the value of variable: ")  
        x = Console.ReadLine()  
    End While  
    Console.WriteLine("程式結束")  
    Console.Read()  
End Sub
```

- ▶ 當 $x \geq 0$ 成立時，就執行迴圈內敘述，不成立時就終止迴圈。
- ▶ **while(條件)** 或 **while 條件** 都可以，即括符可加可不加。



## Ch4. while迴圈指令

---

- ▶ 求最大公約數(Greatest Common Divisor，簡寫G.C.D.，或稱最大公因數)。
- ▶ 指能夠整除多個整數的最大正整數。而多個整數不能都為零。例如8和12的最大公因數為4。
- ▶ 輾轉相除法：兩數相除，取餘數重複進行相除，直到餘數為0時，前一個除數即為最大公約數。



## Ch4. while迴圈指令

### ► 流程圖：

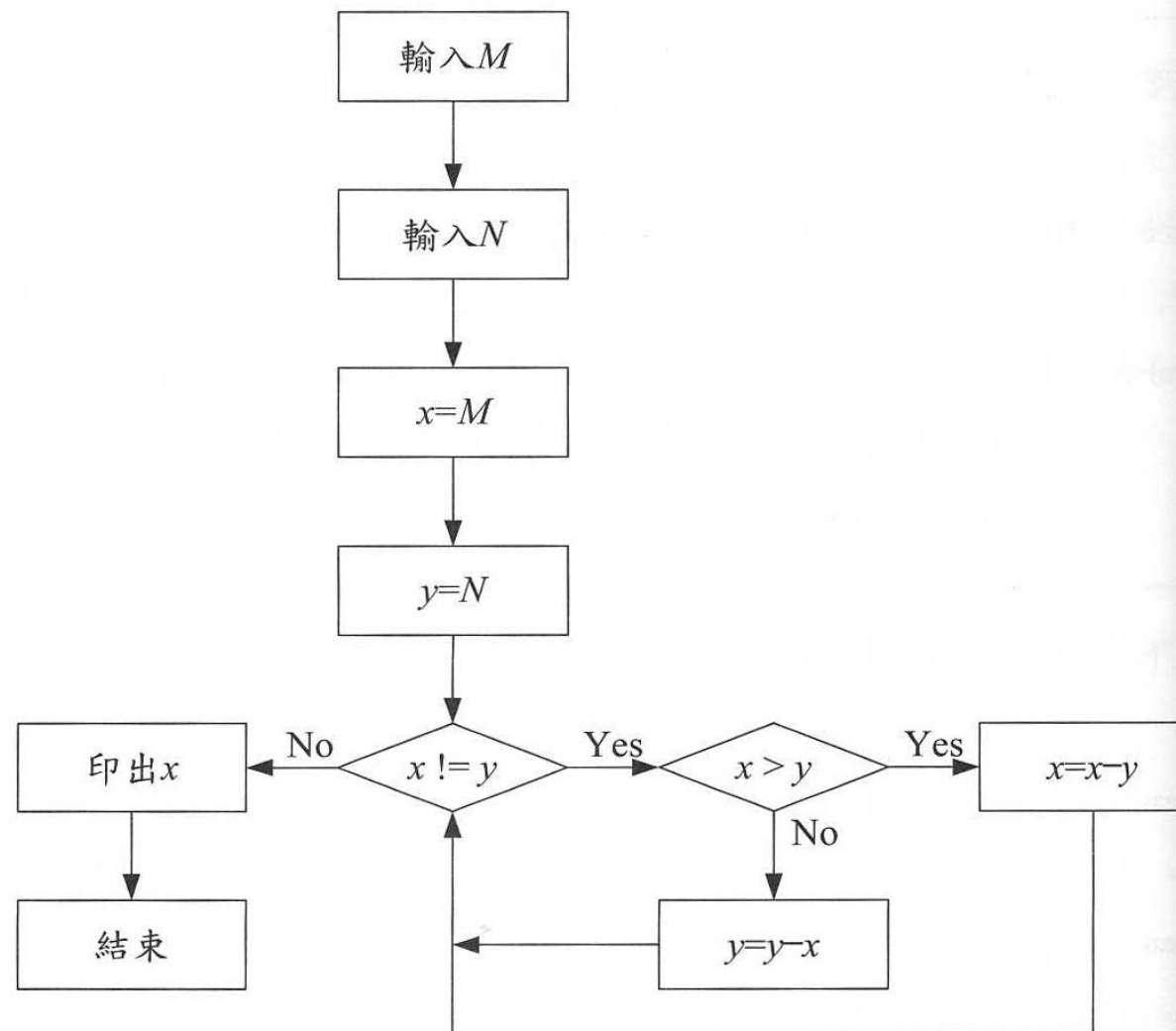


圖 4.2 ► 例題 4.2 的流程圖(求最大公約數)

## Ch4. while迴圈指令

---

### ▶ 參考程式：

```
Sub Main()  
    Dim M, N, x, y As Integer  
    Console.Write("Please input the value of variable M: ")  
    M = Console.ReadLine()  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    x = M : y = N  
    While x <> y  
        If x > y Then  
            x = x - y  
        Else  
            y = y - x  
        End If  
    End While  
    Console.WriteLine("GCD = " & x)  
    Console.Read()  
End Sub
```



## Ch4. while迴圈指令

---

### ▶ 參考程式二：

```
Sub Main()  
    Dim a, b, r, tmp As Integer  
    Console.Write("Please input the value of variable a: ")  
    a = Console.ReadLine()  
    Console.Write("Please input the value of variable b: ")  
    b = Console.ReadLine()  
    If a < b Then      '如果 a < b，把a和b對調  
        tmp = a : a = b : b = tmp  
    End If  
    r = a Mod b  
    While r <> 0      'While迴圈會計算到r=0才會停止  
        a = b : b = r : r = a Mod b  
    End While  
    Console.WriteLine("GCD = " & b)  
    Console.Read()  
End Sub
```



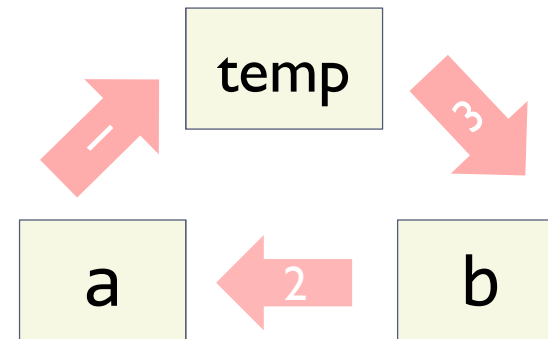


## Ch4. while迴圈指令

---

- ▶ 關於兩變數交換內容的方式：
- ▶ 須借助第三個變數當中介，例如要交換a,b的內容，則需寫成

```
temp = a  
a = b  
b = temp
```



## Ch4. while迴圈指令

- ▶ 計算 $N$ 個數字的和。
- ▶ 檢查 $i$ 有沒有超過 $N$ ，  
超過就不做了。
- ▶ 流程圖：

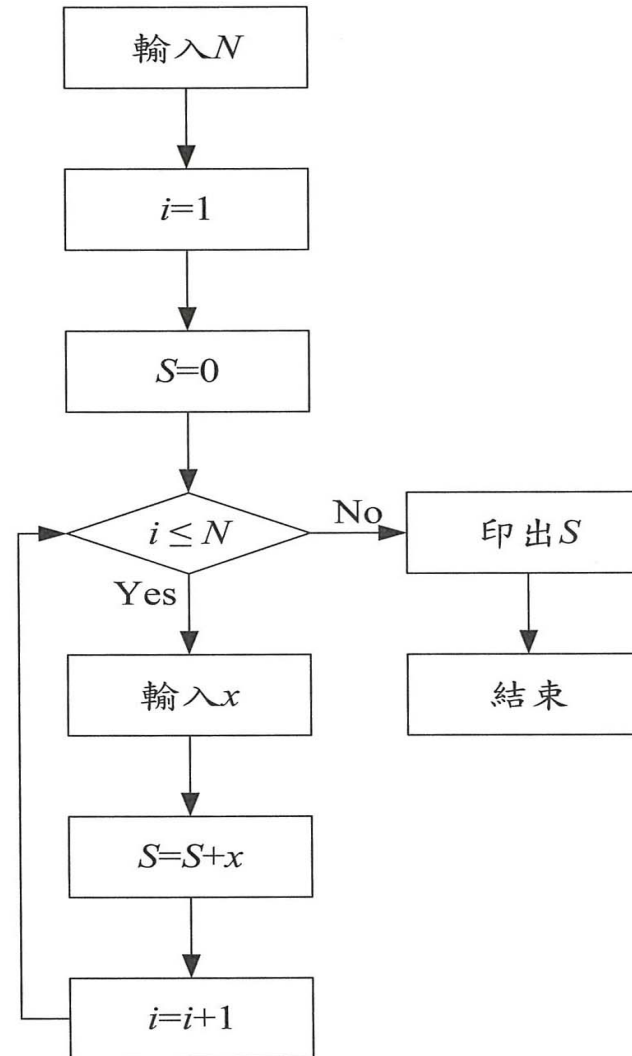


圖 4.3 ▶ 例題 4.3 的流程圖(求  $N$  個數字的和)

## Ch4. while迴圈指令

---

### ▶ 參考程式：

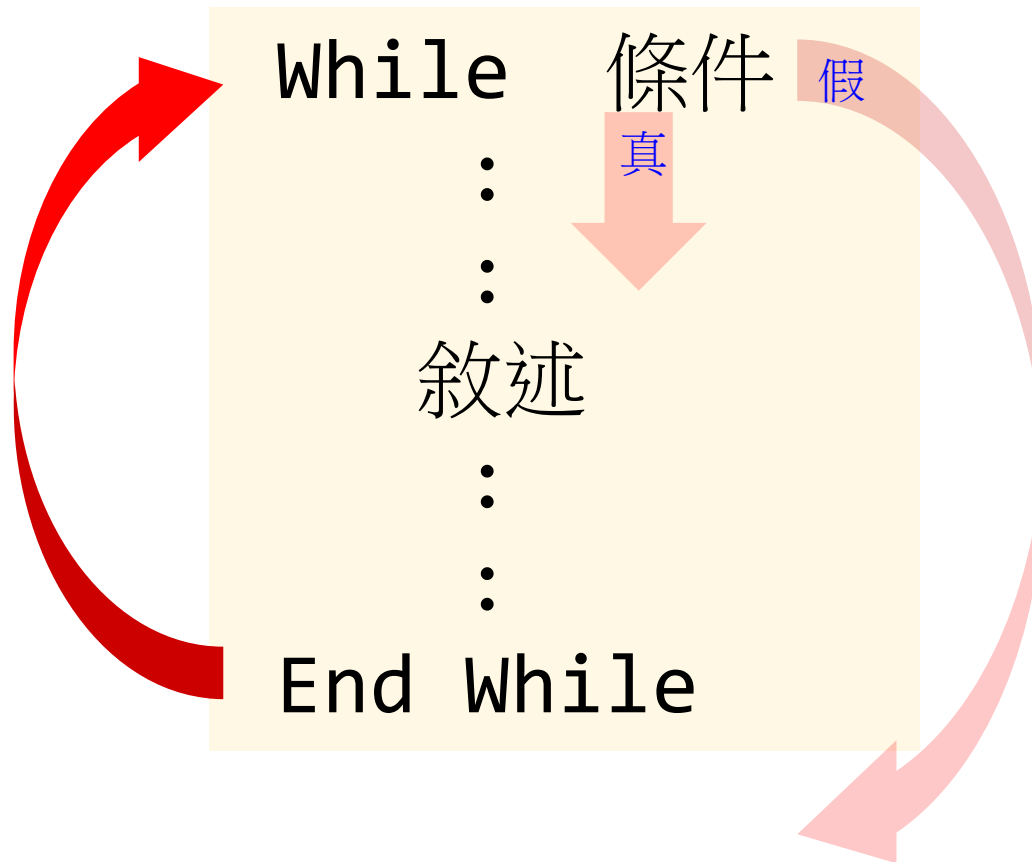
```
Sub Main()  
    Dim N, i, S, x As Integer  
    Console.Write("Please input the value of variable N: ")  
    N = Console.ReadLine()  
    i = 1 : S = 0  
    While i <= N  
        Console.Write("Please input the value of variable x: ")  
        x = Console.ReadLine()  
        S = S + x  
        i = i + 1  
    End While  
    Console.WriteLine("S = " & S)  
    Console.Read()  
End Sub
```



# While ... End While敘述

---

- ▶ 當條件為真時執行迴圈，為假時結束迴圈



# While ... End While敘述

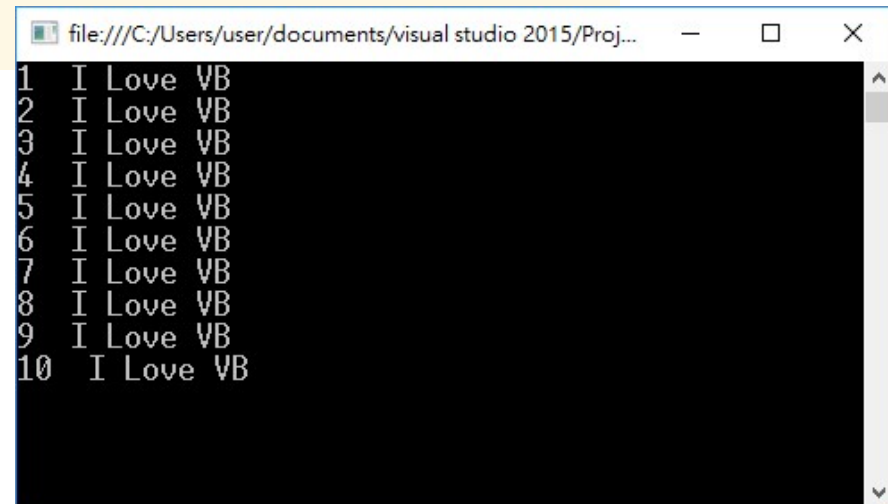
- ▶ 範例練習，印出10次"I Love VB"：

```
Dim i As Integer
i = 1
While i <= 10
    Console.WriteLine(i & " I Love VB")
    i += 1
End While
Console.Read()
```

每次執行條件測試，看是否為真

增/減量

- ▶ 執行結果：



```
1 I Love VB
2 I Love VB
3 I Love VB
4 I Love VB
5 I Love VB
6 I Love VB
7 I Love VB
8 I Love VB
9 I Love VB
10 I Love VB
```

# While ... End While敘述

- ▶ 無盡迴圈，因無意或故意，讓迴圈永遠不會結束
- ▶ 例：

```
While 1=1  
:  
:  
End While
```

故意的，因為  
測試條件永遠  
是真

```
While 1  
:  
:  
End While
```

```
i = 1  
While i <= 10  
    Console.WriteLine(i)  
    ?  
End While
```

無意的，因為忘了改變i值，使  
測試條件永遠是真

## While ... End While敘述

---

- ▶ While當無盡迴圈用是常見的
- ▶ 不是的話記得要有敘述改變測試條件

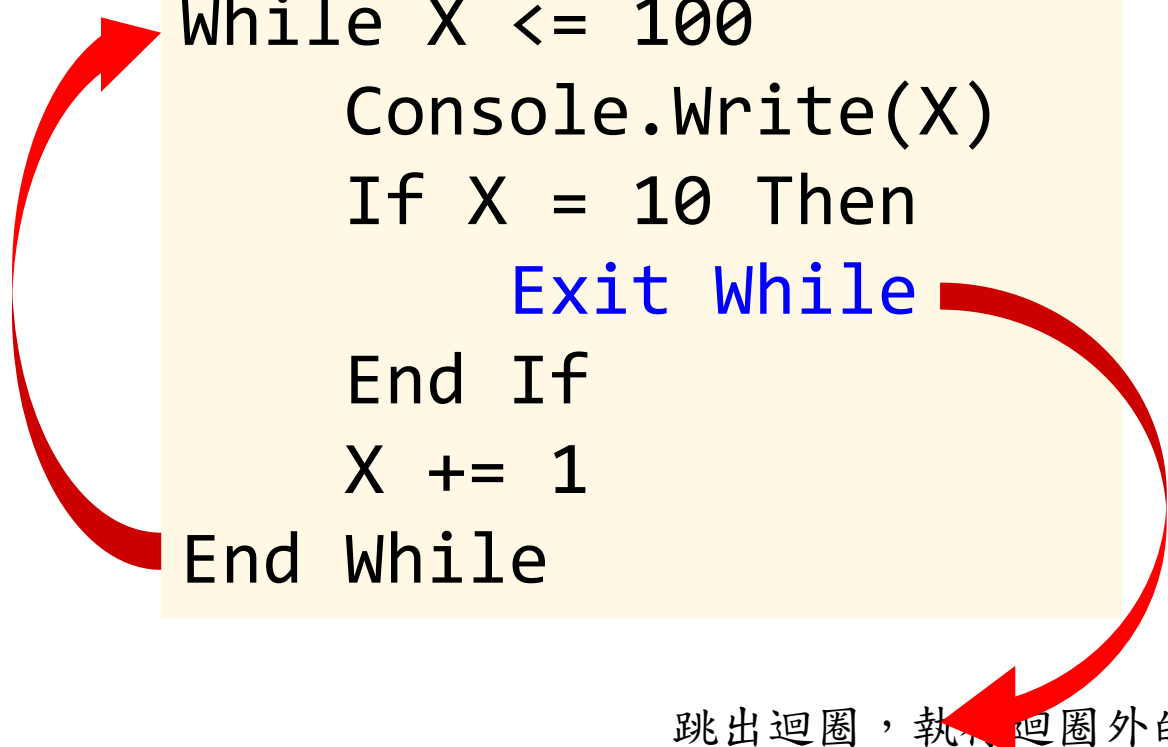
```
While 測試條件  
  :  
  敘述  
  :  
End While
```

← 控制迴圈要不要  
繼續的關鍵

## Exit While敘述

---

- ▶ 此敘述可以強迫跳出所在的迴圈，不執行剩下的敘述，功能與Exit For相同。
- ▶ 範例：



```
While X <= 100
    Console.Write(X)
    If X = 10 Then
        Exit While
    End If
    X += 1
End While
```

跳出迴圈，執行迴圈外的下一道敘述



## Ch4. 習題

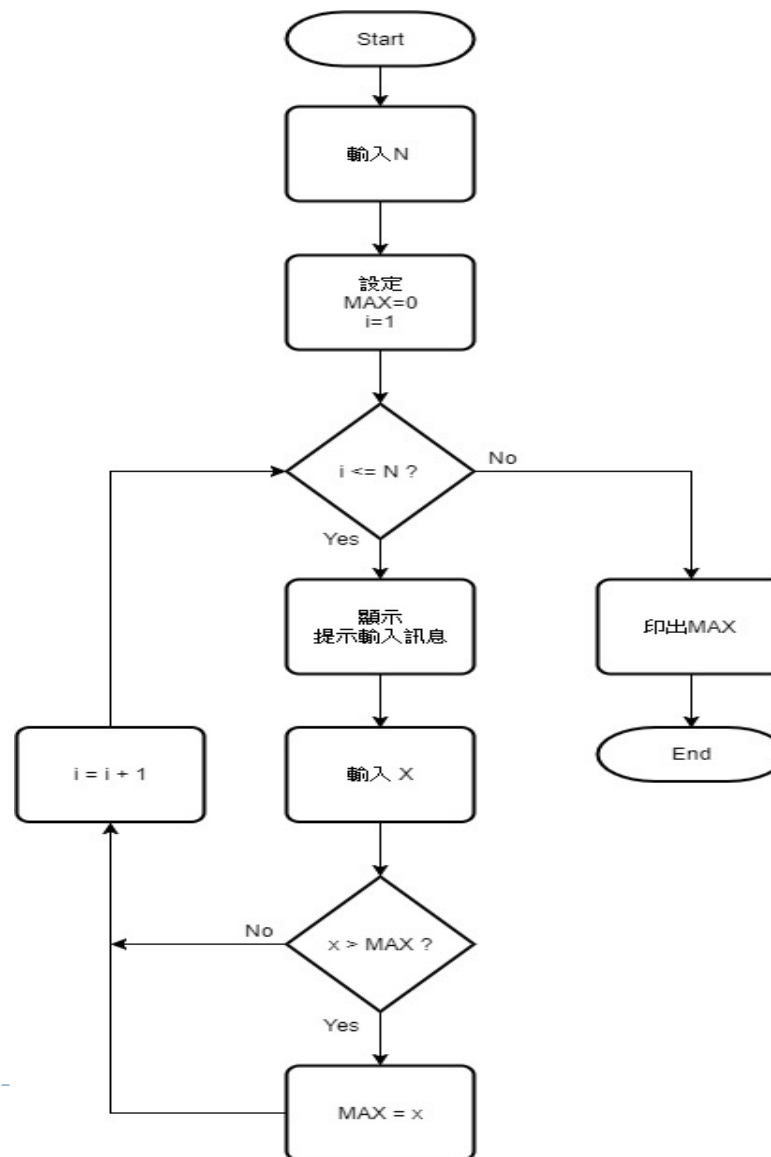
---

- ▶ 4-1. 利用while寫一程式求N個數字的最大值。
- ▶ 4-2. 利用while寫一程式求一個等差級數數字的和，一共有N個數字，程式應該輸入最小的起始值以及數字間的差。(即從起始值開始，間格差，共N個 的總和)
- ▶ 4-3. 利用while寫一程式，讀入N個數字，然後找出所有小於13的數，再求這些數字的和。



## Ch4. 習題

### ► 4-1. 流程圖：



## Ch4. 習題

---

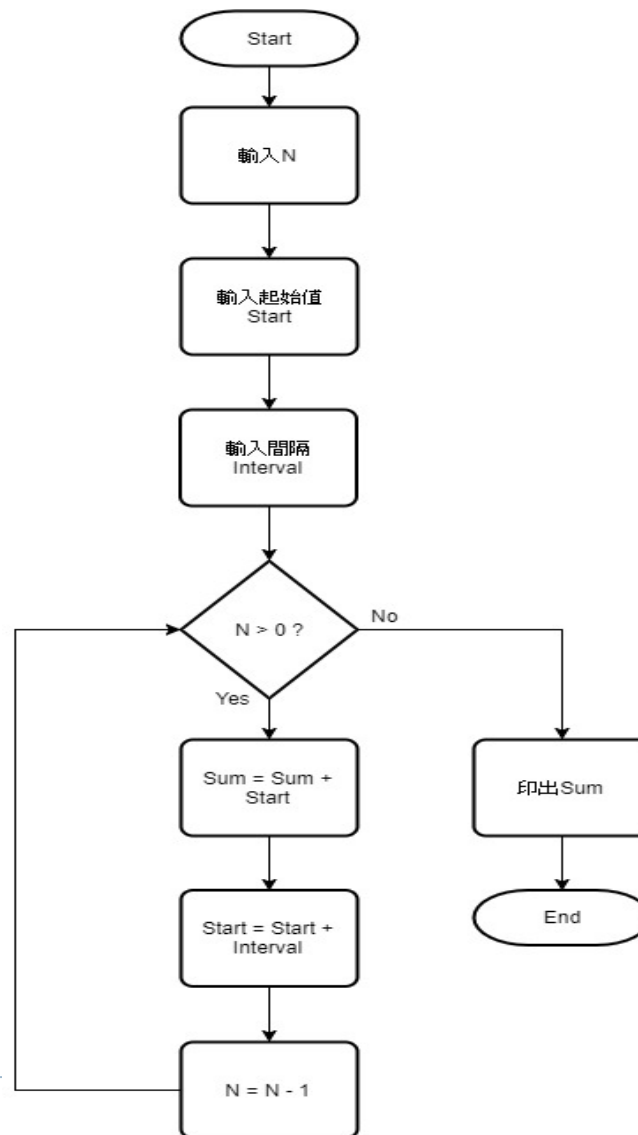
### ► 4-1. 參考程式：

```
Sub Main()  
    Dim N, MAX, i, x As Integer  
    Console.Write("How many number? ")  
    N = Console.ReadLine()  
    MAX = 0 : i = 1  
    While i <= N  
        Console.Write("Please input the value of number" & i & ": ")  
        x = Console.ReadLine()  
        If x > MAX Then  
            MAX = x  
        End If  
        i = i + 1  
    End While  
    Console.WriteLine("MAX = " & MAX)  
    Console.Read()  
End Sub
```



## Ch4. 習題

### ► 4-2. 流程圖：



## Ch4. 習題

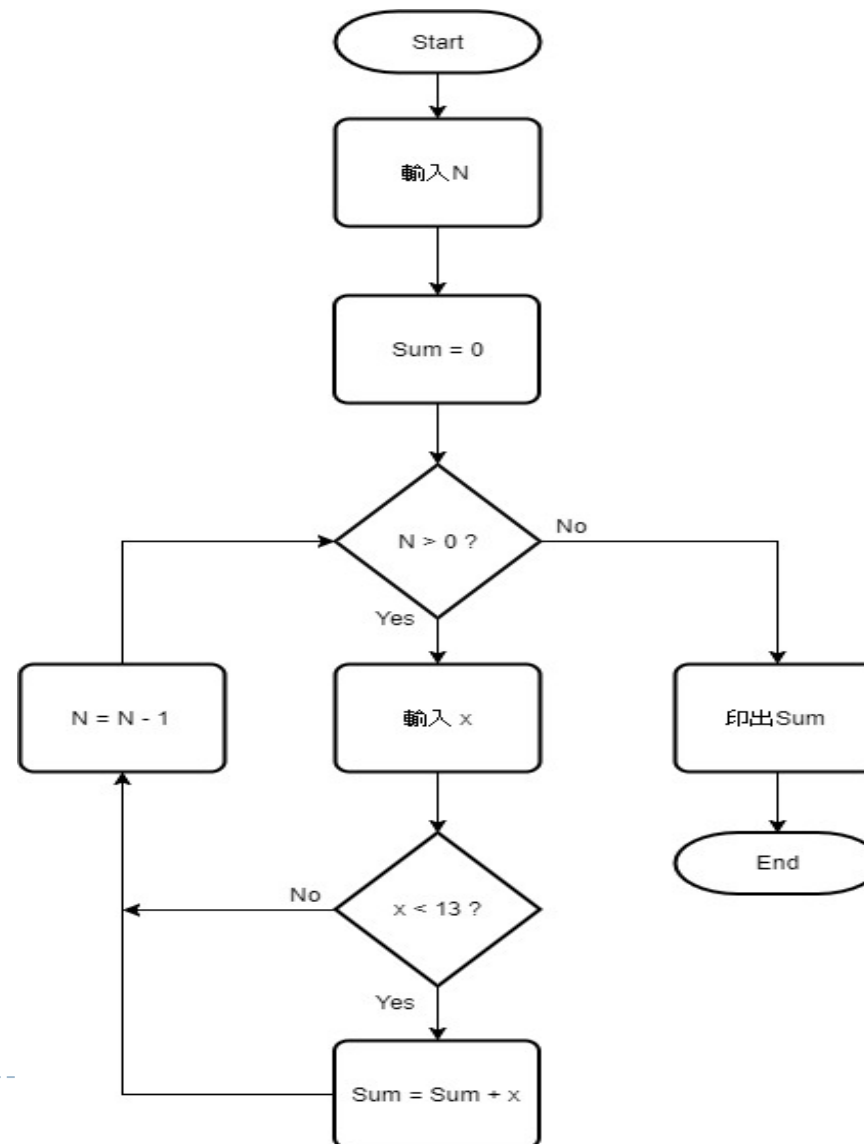
---

### ► 4-2. 參考程式：

```
Sub Main()  
    Dim N, Start, Interval, Sum As Integer  
    Console.Write("Please input N: ")  
    N = Console.ReadLine()  
    Console.Write("Please input the start number: ")  
    Start = Console.ReadLine()  
    Console.Write("Please input the interval: ")  
    Interval = Console.ReadLine()  
    Sum = 0  
    While N > 0  
        Sum += Start  
        Start += Interval  
        N -= 1  
    End While  
    Console.WriteLine("Total = " & Sum)  
    Console.Read()  
End Sub
```

## Ch4. 習題

### ► 4-3. 流程圖：



## Ch4. 習題

---

### ► 4-3. 參考程式：

```
Sub Main()  
    Dim N, Sum, x As Integer  
    Console.Write("Please input N: ")  
    N = Console.ReadLine()  
    Sum = 0  
    While N > 0  
        Console.Write("Please input the value: ")  
        x = Console.ReadLine()  
        If x < 13 Then  
            Sum += x  
        End If  
        N -= 1  
    End While  
    Console.WriteLine("Total = " & Sum)  
    Console.Read()  
End Sub
```



## Ch4. 習題

---

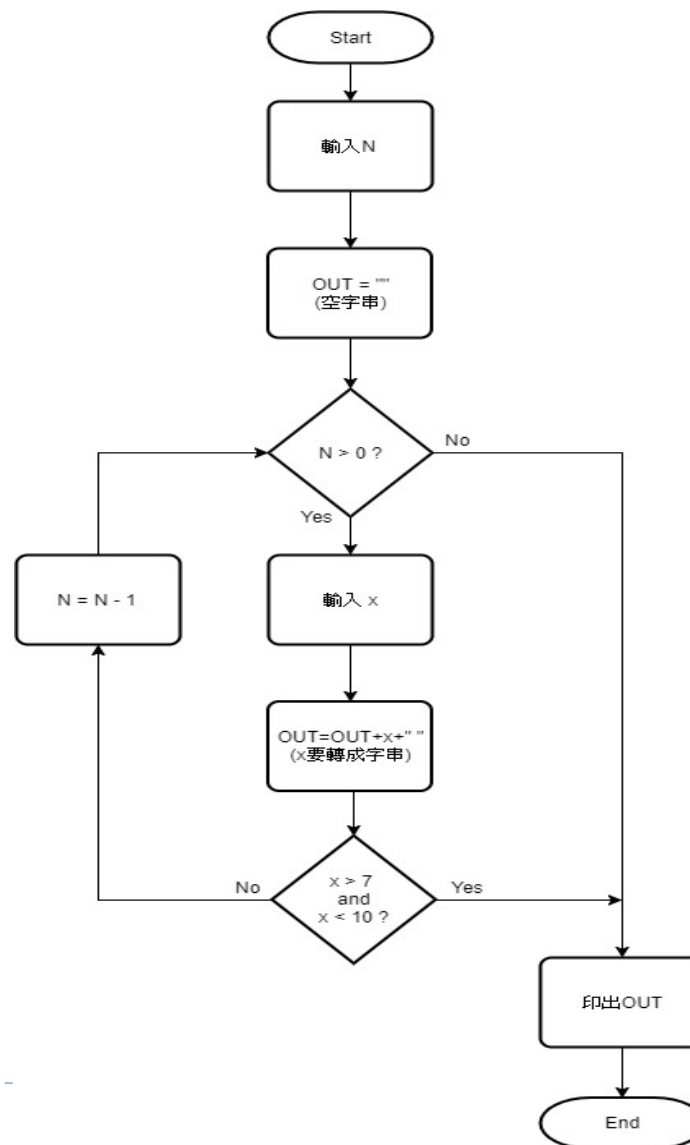
- ▶ 4-4. 利用while寫一程式，讀入N個數字，找到第一個大於7而小於10的數字就停止，而且列印出這個數字。
- ▶ 4-5. 利用while寫一程式，讀入 $a_1, a_2, \dots, a_5$ 和 $b_1, b_2, \dots, b_5$ 。找到第一個 $a_i > b_i$ ，即停止，並列印出 $a_i$ 及 $b_i$ 。





## Ch4. 習題

### ► 4-4. 流程圖：



## Ch4. 習題

---

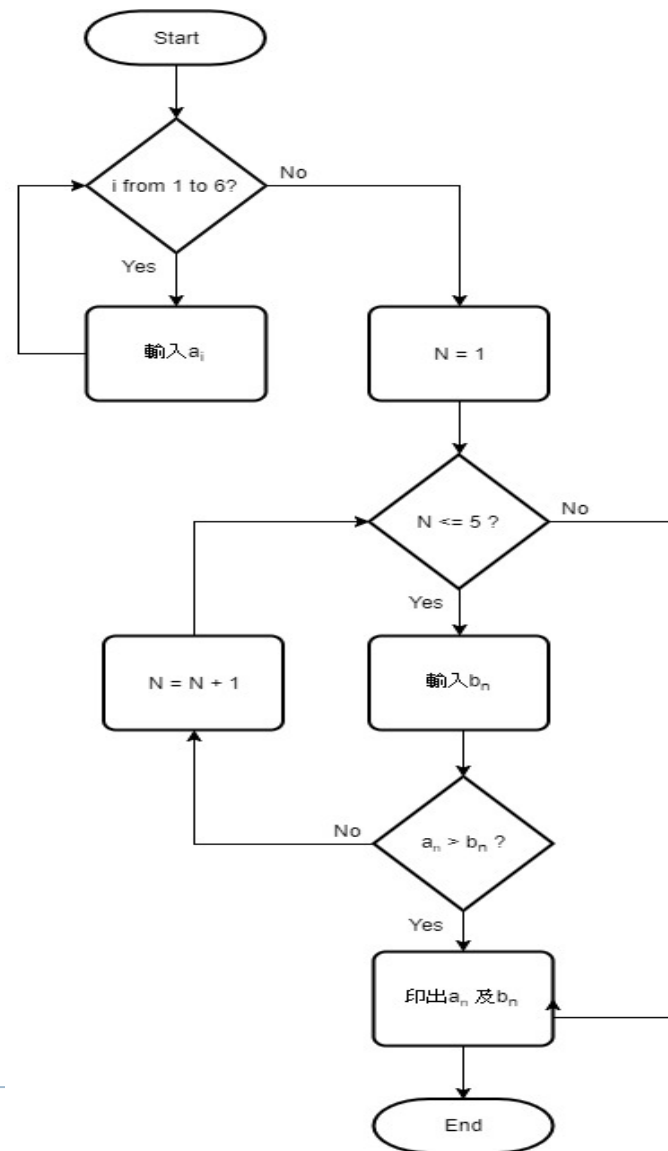
### ► 4-4. 參考程式：

```
Sub Main()  
    Dim N, x As Integer  
    Console.Write("Please input N: ")  
    N = Console.ReadLine()  
    While N > 0  
        Console.Write("Please input the value: ")  
        x = Console.ReadLine()  
        If x > 7 And x < 10 Then  
            Exit While  
        End If  
        N -= 1  
    End While  
    Console.WriteLine("STOP ON: " & x)  
    Console.Read()  
End Sub
```



## Ch4. 習題

### ► 4-5. 流程圖：



## Ch4. 習題

---

### ► 4-5. 參考程式(依序輸入a和b)

```
Sub Main()  
    Dim N, a, b As Integer  
    N = 1  
    While N <= 5  
        Console.Write("Please input a" & N & ": ")  
        a = Console.ReadLine()  
        Console.Write("Please input b" & N & ": ")  
        b = Console.ReadLine()  
        If a > b Then  
            Console.WriteLine("a" & N & " = " & a & " > b" & N & " = " & b)  
            Exit While  
        End If  
        N += 1  
    End While  
    Console.Read()  
End Sub
```



## Ch4. 習題

- 4-5. 參考程式(用陣列來做，但進度在下一章)：

```
Sub Main()  
    Dim N, a(5), b(5) As Integer  
    For i = 1 To 5 '先輸入a的五個值，存入陣列  
        Console.Write("Please input a" & i & ": ")  
        a(i) = Console.ReadLine()  
    Next  
    N = 1  
    While N <= 5 '輸入b的五個值，並做比較  
        Console.Write("Please input b" & N & ": ")  
        b(N) = Console.ReadLine()  
        If a(N) > b(N) Then  
            Console.WriteLine("a" & N & " = " & a(N) &  
                                " > b" & N & " = " & b(N))  
        End If  
        Exit While  
        N += 1  
    End While  
    Console.Read()  
End Sub
```

# Do ... Loop敘述

---

- ▶ 還有與While相似的敘述：Do - Loop
- ▶ 區分為前測式與後測式迴圈：

- ▶ 前測式

先判斷條件再執行迴圈

```
Do While 條件
  :
  :
Loop
```

```
Do Until 條件
  :
  :
Loop
```

- ▶ 後測式

先執行迴圈再判斷條件

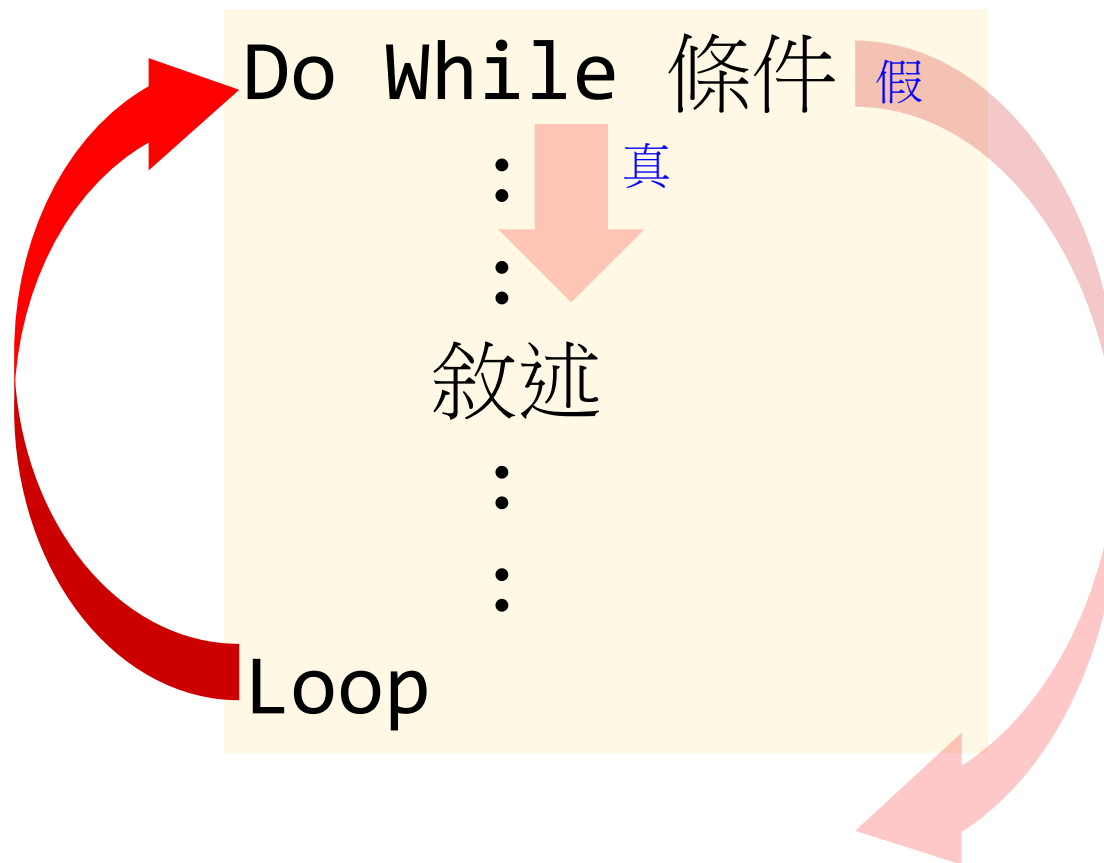
```
Do
  :
  :
Loop While 條件
```

```
Do
  :
  :
Loop Until 條件
```

## Do ... Loop敘述

---

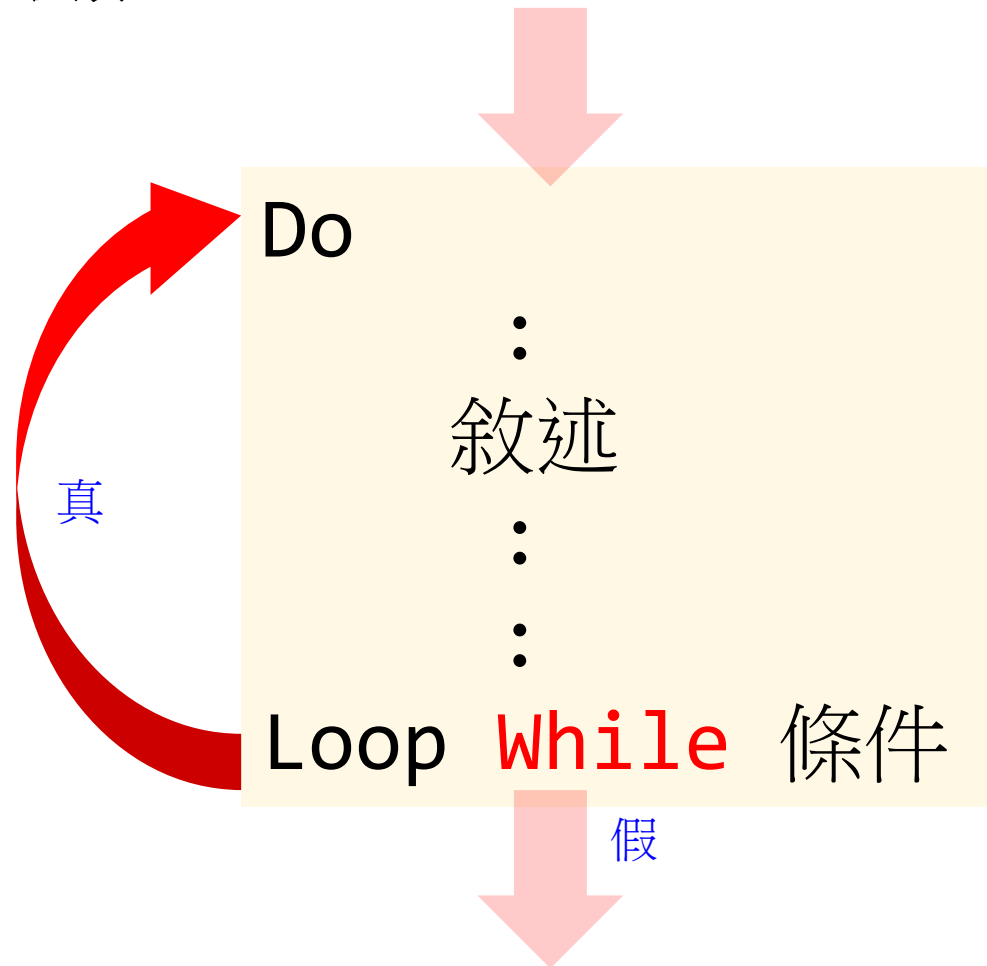
- ▶ 前測式效果與While敘述相同，端看各位喜歡用哪種敘述來寫



# Do ... Loop敘述

---

- ▶ 後測式語法While :

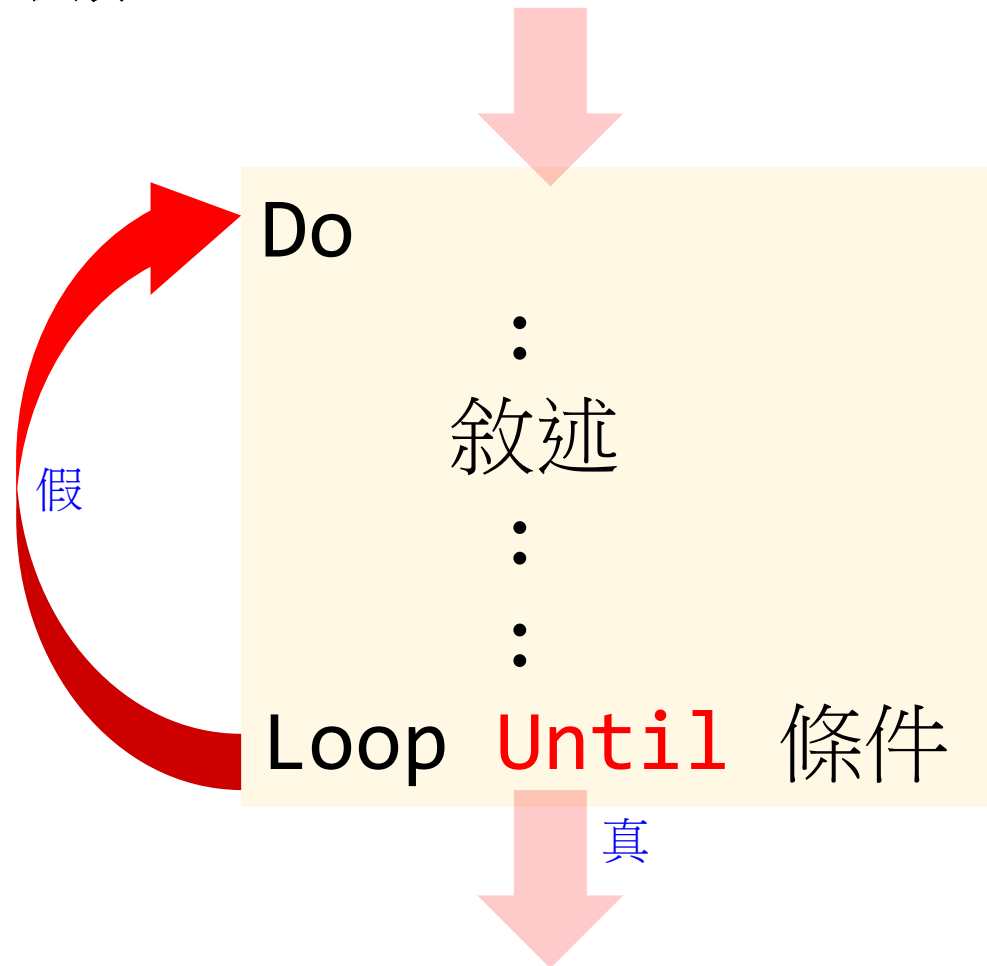




# Do ... Loop敘述

---

- ▶ 後測式語法Until：

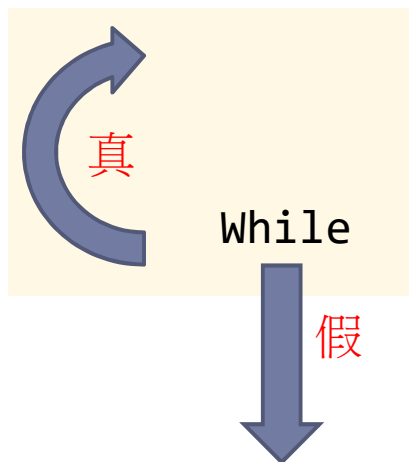


# Do ... Loop敘述

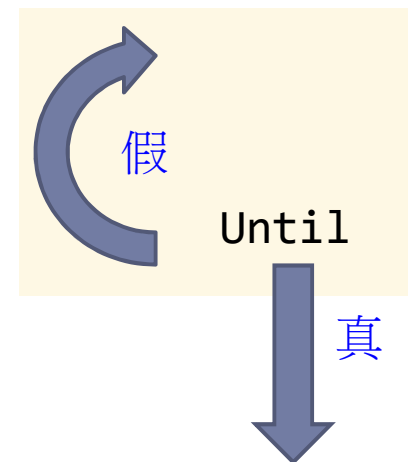
---

► 記住：

**While**是條件  
為真時要重複  
為假時要離開

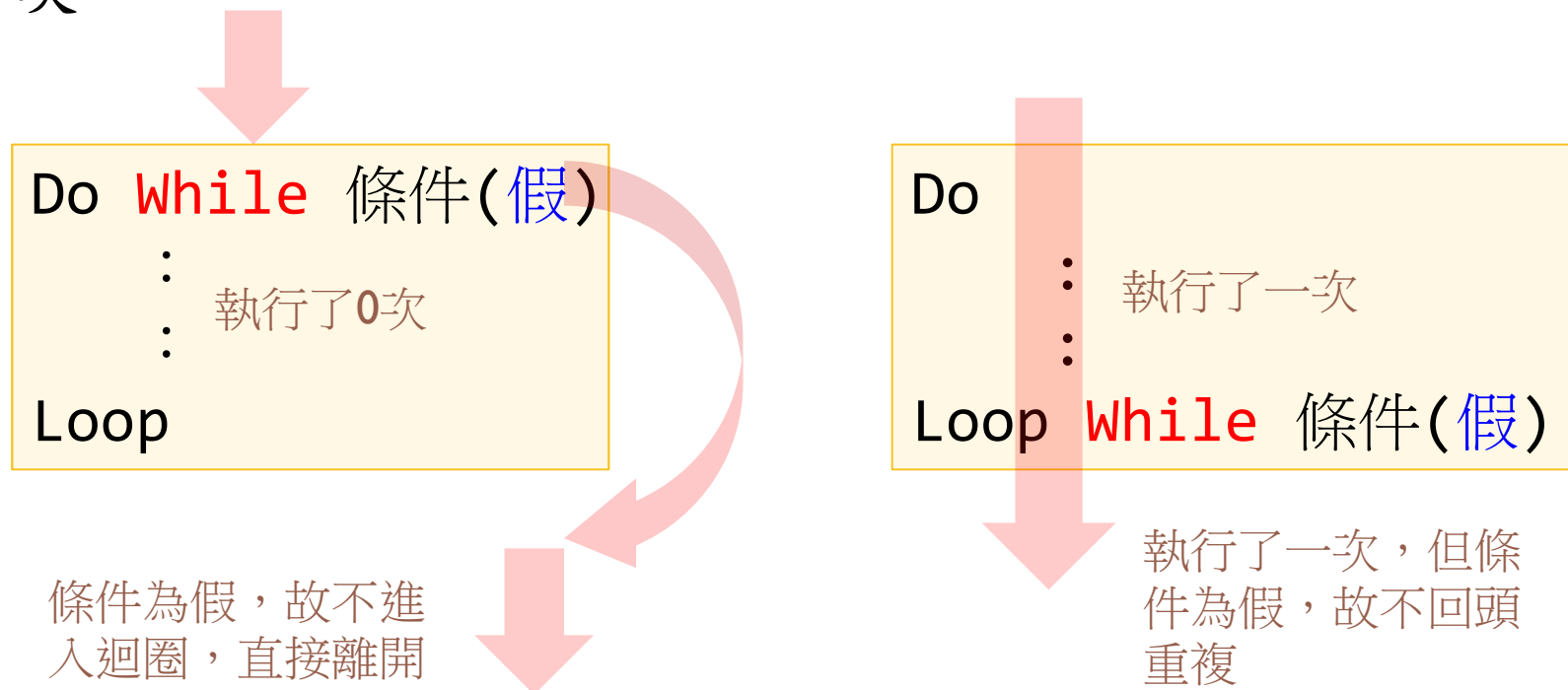


**Until**是條件  
為假時要重複  
為真時要離開



## Do ... Loop敘述

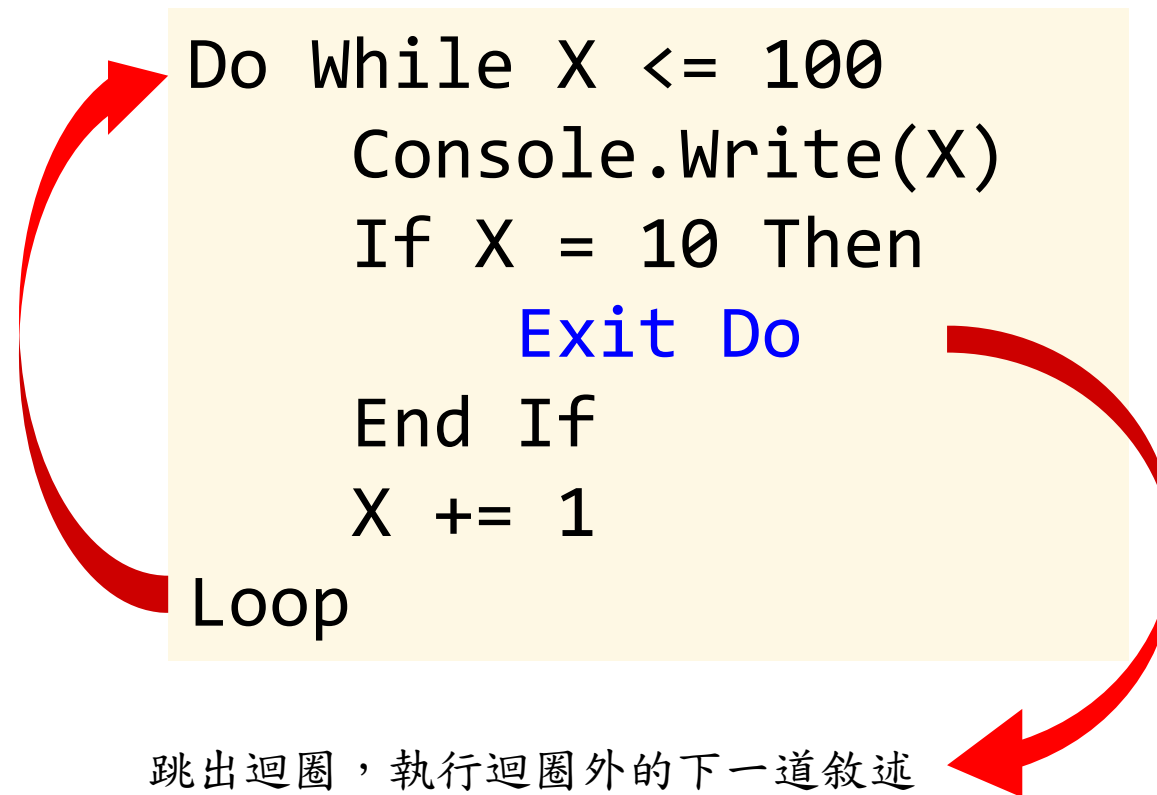
- ▶ 後測式是先執行迴圈內的敘述，再判斷要不要繼續重複執行
- ▶ 後測式最大的不同是迴圈內的敘述一定會執行至少一次



## Exit Do敘述

---

- ▶ 此敘述可以強迫跳出所在的迴圈，不執行剩下的敘述，功能與Exit While相同。
- ▶ 範例：



# Continue 指令

---

- ▶ **Continue For :**

跳過For迴圈剩下的指令，回到For的開頭，執行下一個For/Next迴圈。

- ▶ **Continue Do :**

忽略之後的指令，跳回Do去執行。

- ▶ 跟Exit指令相反

## Do Loop練習 參考寫法

---

- ▶ 1. 由鍵盤輸入10個數值並將其加總(前置)

```
Dim i, x, sum As Integer
i = 0 : sum = 0
Do While i < 10
    x = Console.ReadLine()
    sum += x
    i += 1
Loop
Console.WriteLine("Sum= " & sum)
Console.Read()
```

- ▶ 說明：冒號 `:` 是將多行敘述寫在同一行時當分隔符號

休息一下~

---



## Ch5.陣列

---

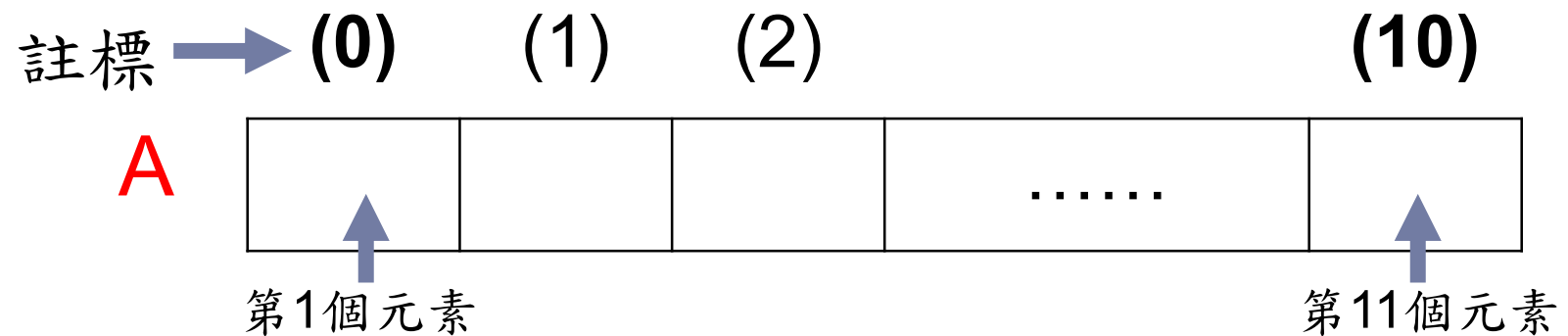
- ▶ 在過去的章節，我們的做法是每次讀入一個資料，就立刻去處理，當讀入的資料一多就很不方便，應該先儲存至記憶體，以便後續處理，最普通的辦法是使用陣列。
- ▶ 目的：
  - ▶ 減少變數的個數,使程式的編寫更容易
  - ▶ 陣列可配合迴圈來處理大量資料





## Ch5.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

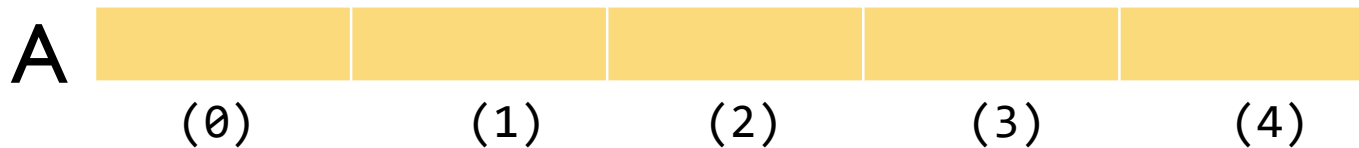
- ▶ 宣告：任何陣列在使用前必須先宣告,格式為

**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

---

- ▶ 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

## Ch5.陣列

- ▶ 利用一維陣列求10個數字的計算平均值。
- ▶ 流程圖：

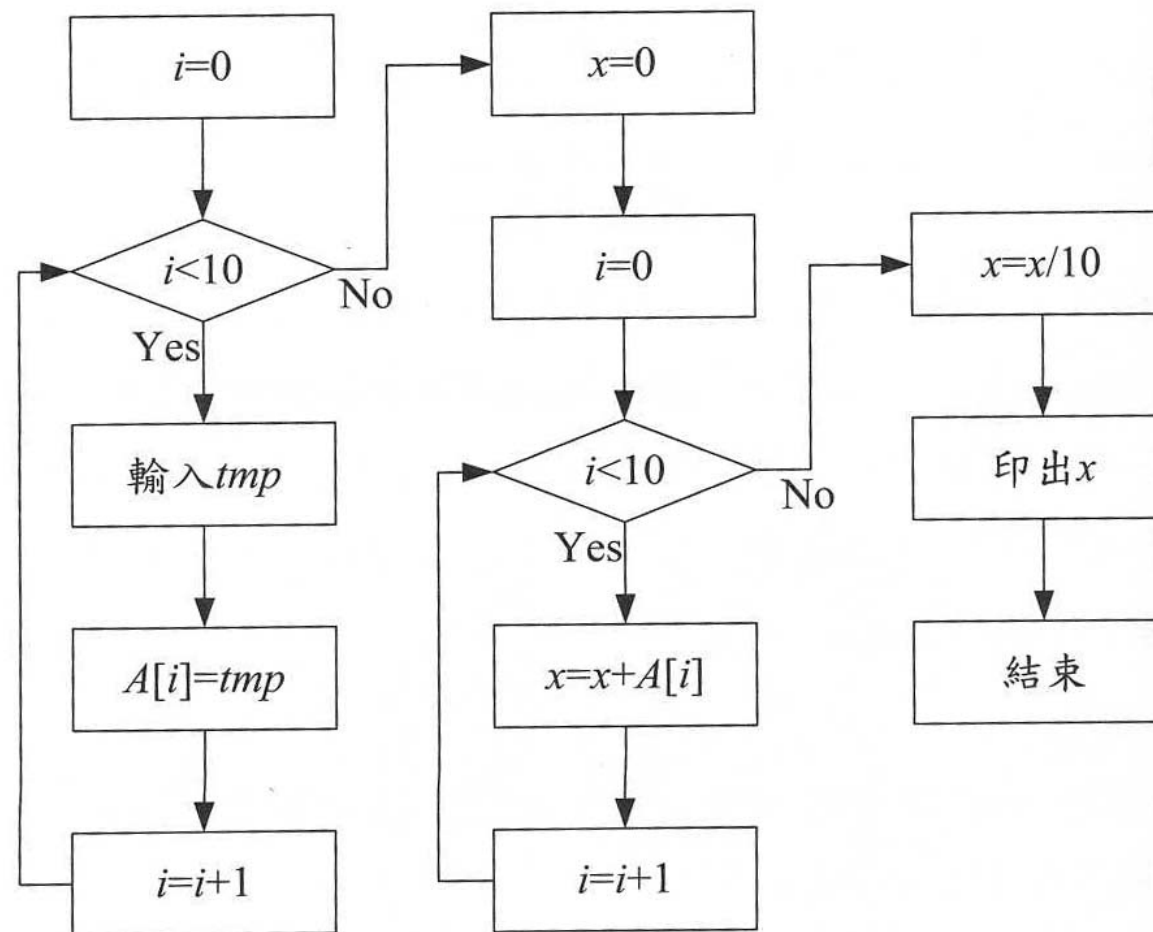


圖 5.3 ▶ 例題 5.1 的流程圖(求一個陣列中 10 個數字的平均數)

## Ch5.陣列

---

### ▶ 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



## Ch5.陣列

► 利用一維陣列求10個數字的最大值。

► 流程圖：

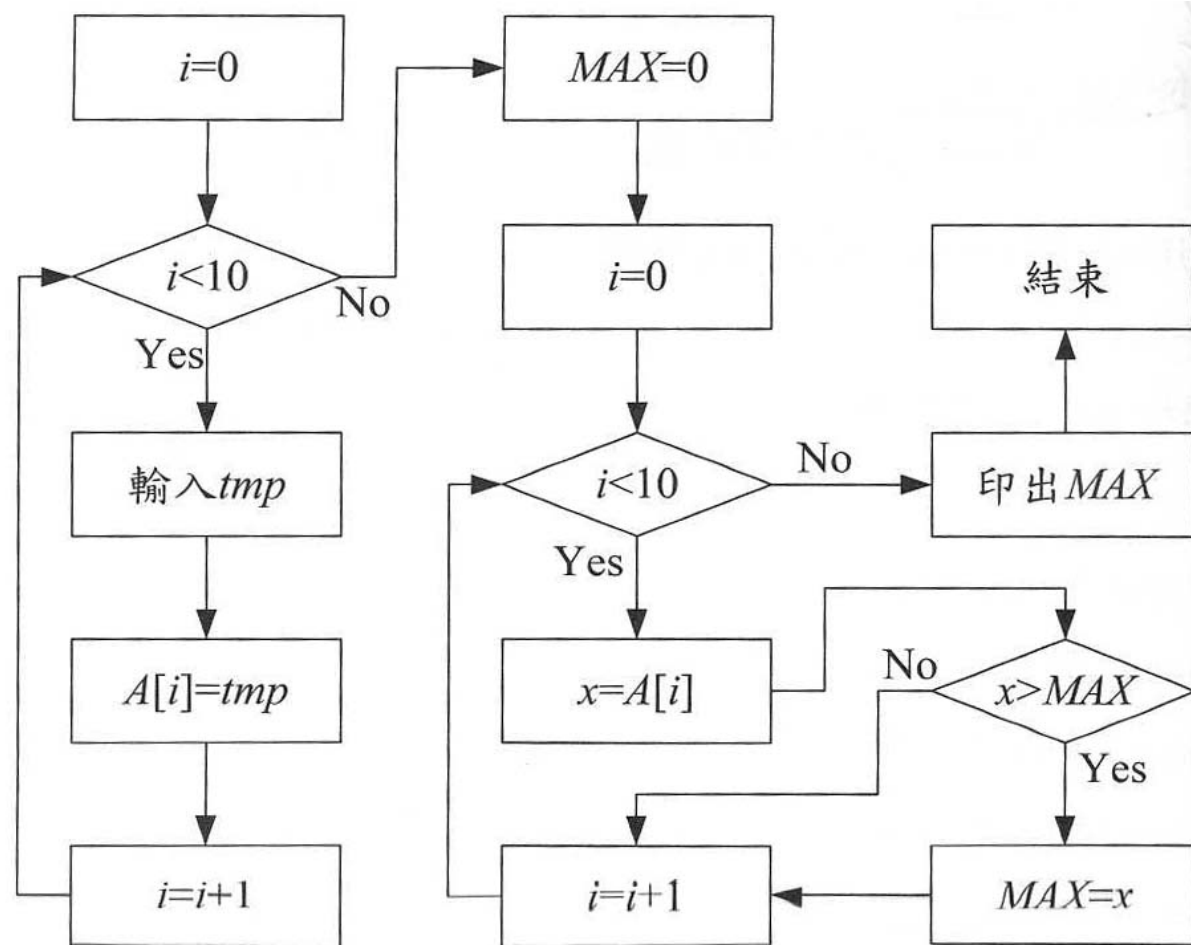


圖 5.4 ► 例題 5.2 的流程圖(求一個陣列中 10 個數字的最大)

## Ch5.陣列

---

### ▶ 參考程式：

```
Sub Main()  
    Dim A(10), MAX, x As Integer    '宣告一維陣列A(10)  
    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  
    MAX = 0  
    For i = 1 To 10  
        x = A(i)  
        If x > MAX Then  
            MAX = x  
        End If  
    Next  
    Console.WriteLine("MAX = " & MAX)  
    Console.Read()  
End Sub
```

## Ch5.陣列

- ▶ 搜尋問題：輸入10個數字至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。
- ▶ 流程圖：

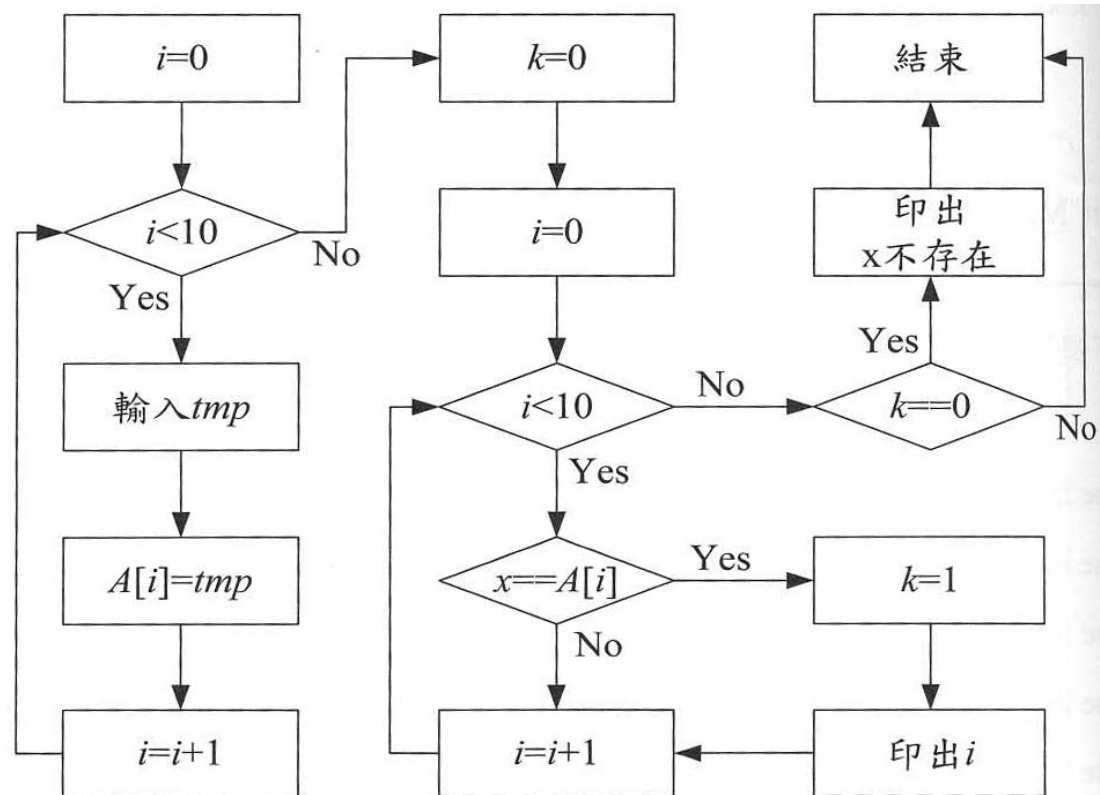


圖 5.5 ▶ 例題 5.3 的流程圖(搜尋問題)



## Ch5.陣列

---

### ▶ 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 you 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
```

## Ch5.陣列

- ▶ 二維陣列：使用兩個註標來宣告並使用陣列。
- ▶ 宣告方式：

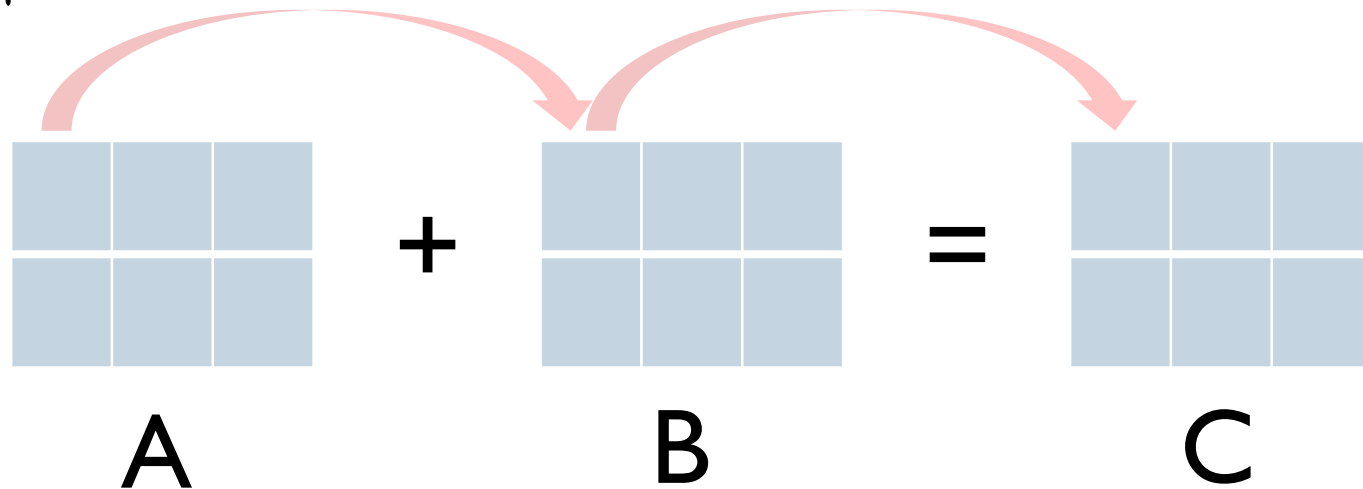
```
Dim A(3,2) As Integer
```

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

## Ch5.陣列

---

- ▶ 兩個二維矩陣相加。以陣列模擬矩陣，求兩個2乘3的二維矩陣相加之結果，第一個及第二個矩陣分別以A及B表示，相加之結果存入C矩陣，最後將C矩陣內容顯示出來。
- ▶ 圖示：



## Ch5.陣列

### ► 流程圖：

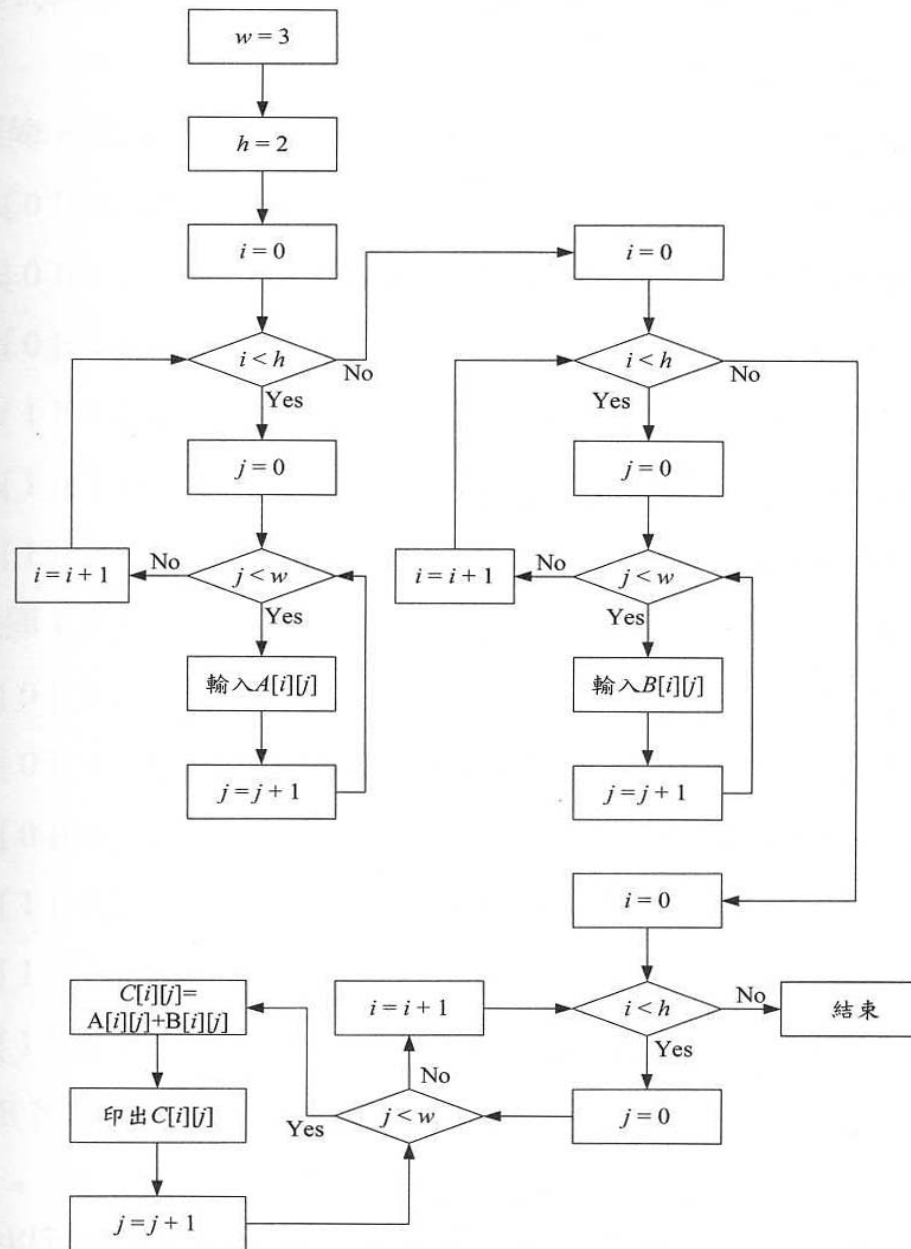


圖 5.6 ► 例題 5.4 的流程圖

## Ch5.陣列

---

▶ 參考程式：

```
Sub Main()  
    Dim h, w As Integer  
    h = 2 : w = 3  
    Dim A(h, w), B(h, w), C(h, w) As Integer '宣告三個二維陣列  
    Console.WriteLine("請輸入矩陣A的值...")  
    For i = 1 To h  
        For j = 1 To w  
            Console.Write("請輸入A(" & i & ")( " & j & ")的值: ")  
            A(i, j) = Console.ReadLine()  
        Next  
    Next  
    Console.WriteLine("請輸入矩陣B的值...")  
    For i = 1 To h  
        For j = 1 To w  
            Console.Write("請輸入B(" & i & ")( " & j & ")的值: ")  
            B(i, j) = Console.ReadLine()  
        Next  
    Next  
Next
```

## Ch5.陣列

---

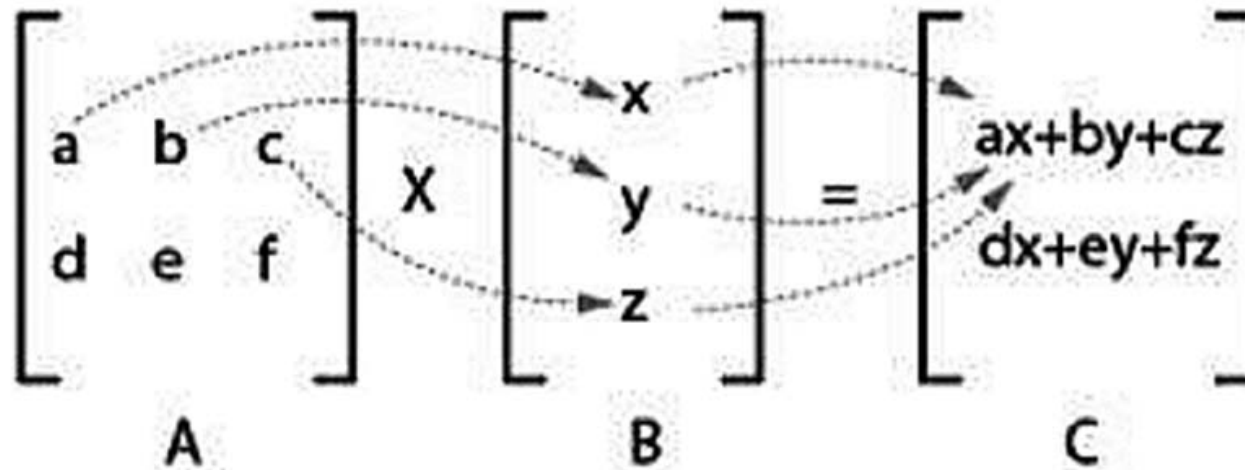
### ▶ 參考程式：

```
Console.WriteLine("矩陣A、B相加結果儲存於矩陣C中：")
For i = 1 To h      '計算A(i,j)+B(i,j)，將結果存入C(i,j)並印出
    For j = 1 To w
        C(i, j) = A(i, j) + B(i, j)
        Console.WriteLine("C(" & i & ", " & j & ") = " & C(i, j))
    Next
Next
Console.Read()
End Sub
```



## Ch5.陣列

- ▶ 兩個矩陣相乘。求2乘3的A矩陣乘以3乘1的B矩陣，結果存入2乘1的C矩陣，最後將C矩陣的內容顯示出來。
- ▶ 圖示：



## Ch5.陣列

---

### ▶ 參考程式：

```
Sub Main()  
    Dim ha, wa, hb, wb As Integer  
    ha = 2 : wa = 3 : hb = 3 : wb = 1  
    '宣告三個二維陣列  
    Dim A(ha, wa), B(hb, wb), C(ha, wb) As Integer  
    Console.WriteLine("請輸入矩陣A的值...")  
    For i = 1 To ha  
        For j = 1 To wa  
            Console.Write("請輸入A(" & i & ")( " & j & ")的值: ")  
            A(i, j) = Console.ReadLine()  
        Next  
    Next  
    Console.WriteLine("請輸入矩陣B的值...")  
    For i = 1 To hb  
        For j = 1 To wb  
            Console.Write("請輸入B(" & i & ")( " & j & ")的值: ")  
            B(i, j) = Console.ReadLine()  
        Next  
    Next  
Next
```



## Ch5.陣列

---

### ▶ 參考程式：

```
For i = 1 To ha
    For j = 1 To wb
        For k = 1 To wa
            '將A(i,k)*B(k,j)的結果累加入C(i,j)
            C(i, j) = C(i, j) + A(i, k) * B(k, j)
        Next
        Console.WriteLine("C(" & i & "," & j & ") = " & C(i, j))
    Next
Next
Console.Read()
End Sub
```



## 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

---

- ▶ 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()
```



## 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...



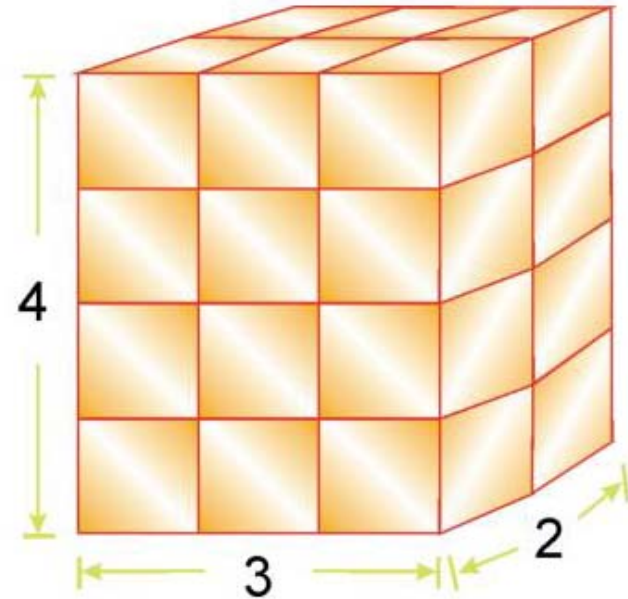
## 三維陣列

### ► 宣告：

Dim 陣列名稱(註標一,註標二,註標三) As 資料型態

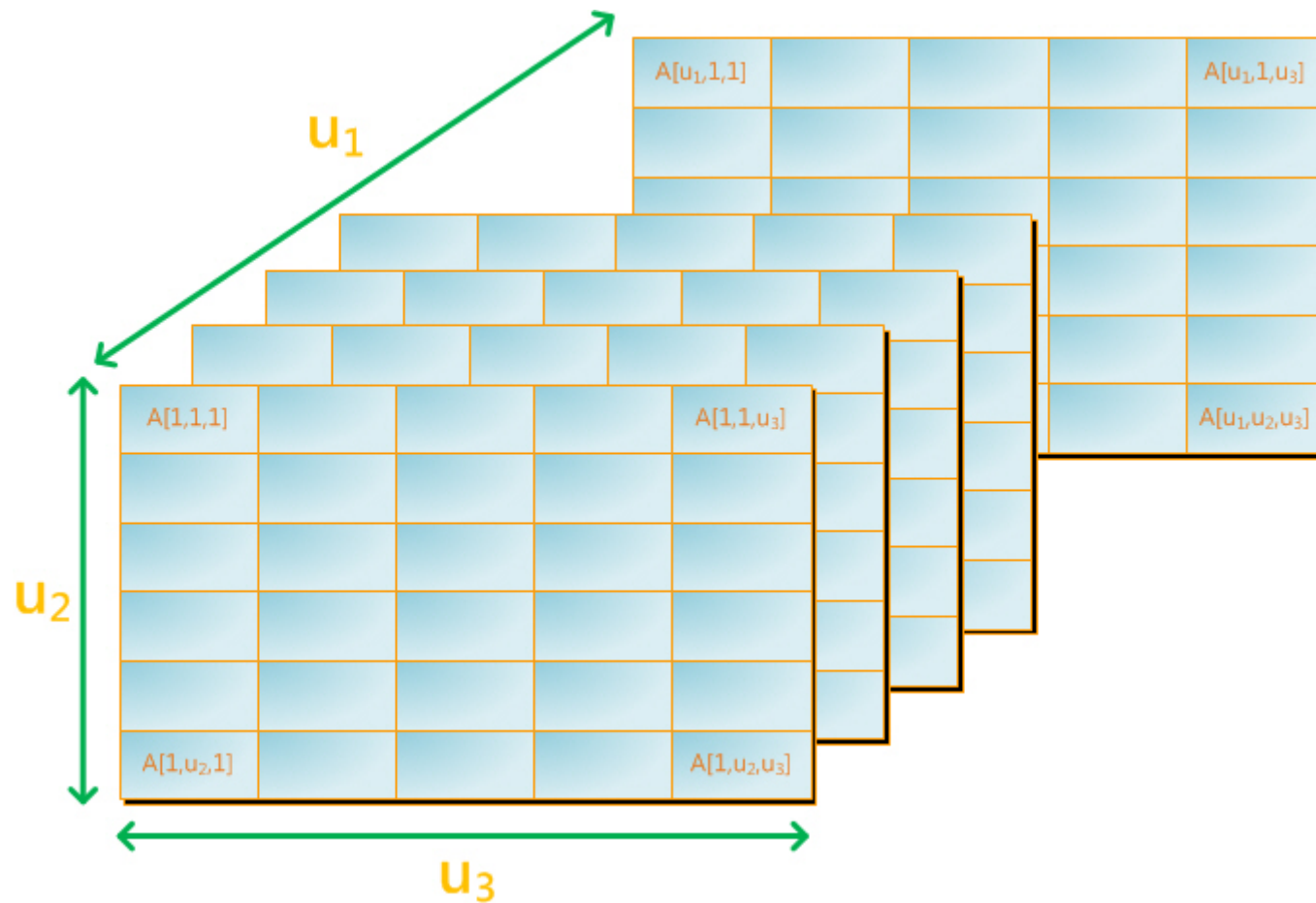
### ► 例如：

Dim A(4,3,2) As Integer



# 三維陣列

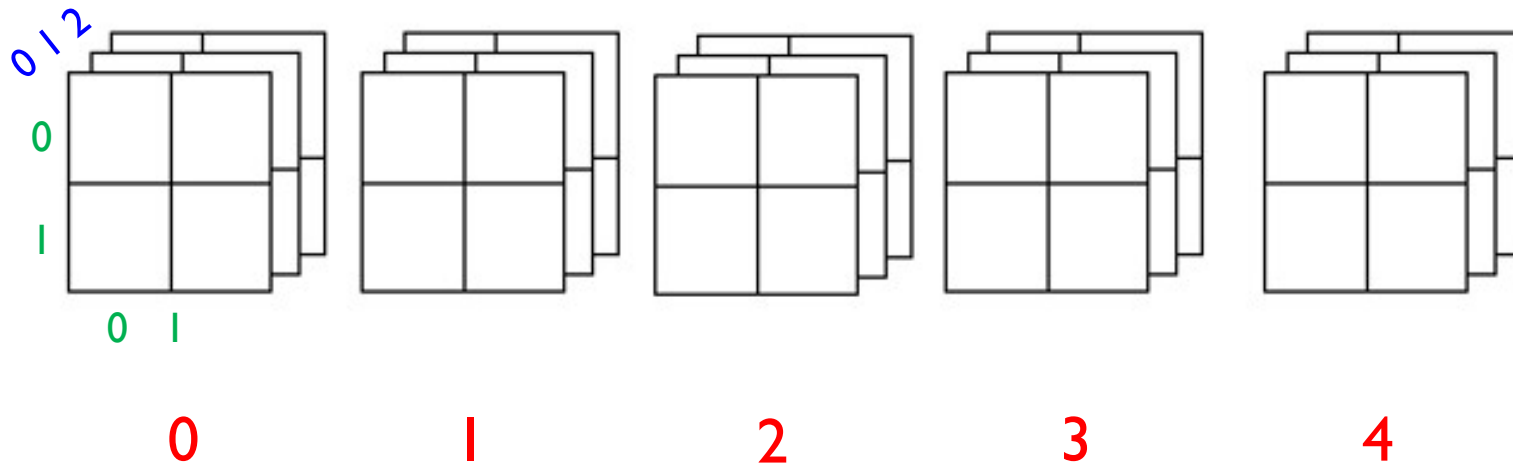
- ▶ 可想成是多個二維陣列



## 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

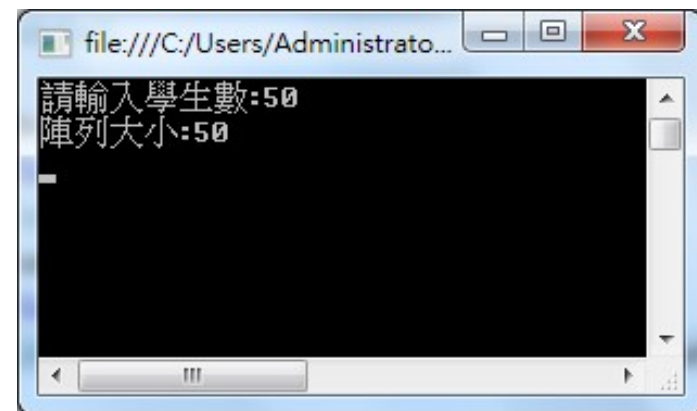
‘在程式中指定變數值

‘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)
```

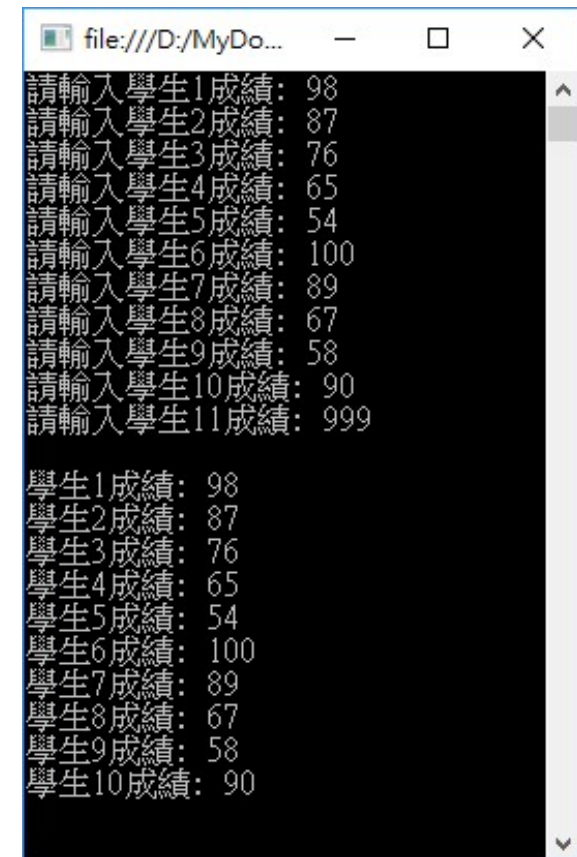
```
ReDim A(1,4,3)
```

‘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

## Ch5. 習題

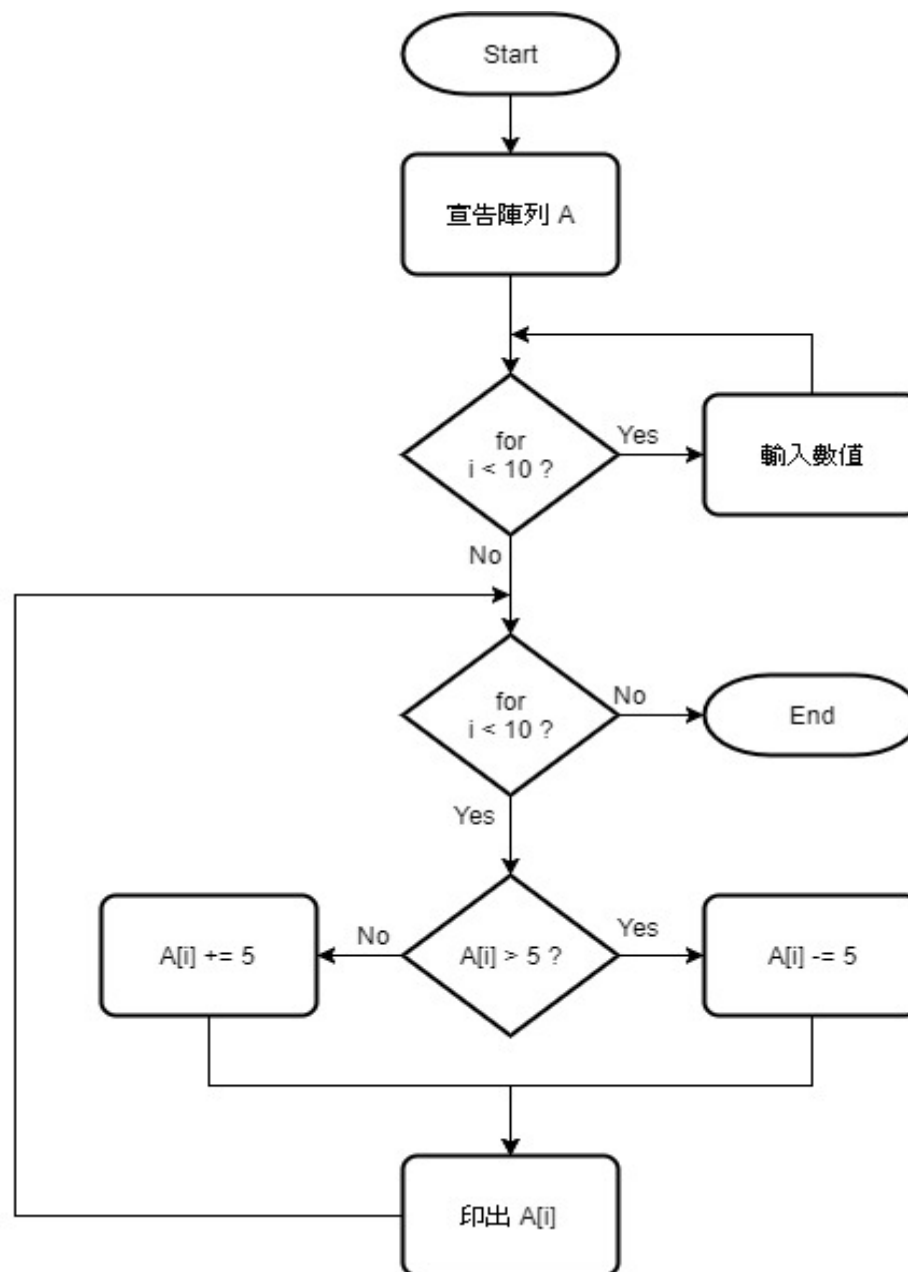
---

- ▶ 5-1. 寫一程式，將10個數字讀入A陣列，然後逐一檢查此陣列，如 $A(i) > 5$ ，則令 $A(i) = A(i) - 5$ ，否則 $A(i) = A(i) + 5$ 。
- ▶ 5-2. 寫一程式，將10個數字讀入A陣列，對每一個數字，令 $A(i) = A(i) + i$ 。
- ▶ 5-3. 寫一程式，將10個數字讀入A陣列，並建立一個B陣列，如 $A(i) \geq 0$ ，令 $B(i) = 1$ ，否則令 $B(i) = 0$ 。
- ▶ 5-4. 寫一程式，將15數字存入 $3 \times 5$ 的二維陣列A中，求每一行及每一列數字的和。



## Ch5. 習題

### ► 5-1 流程圖：



## Ch5. 習題

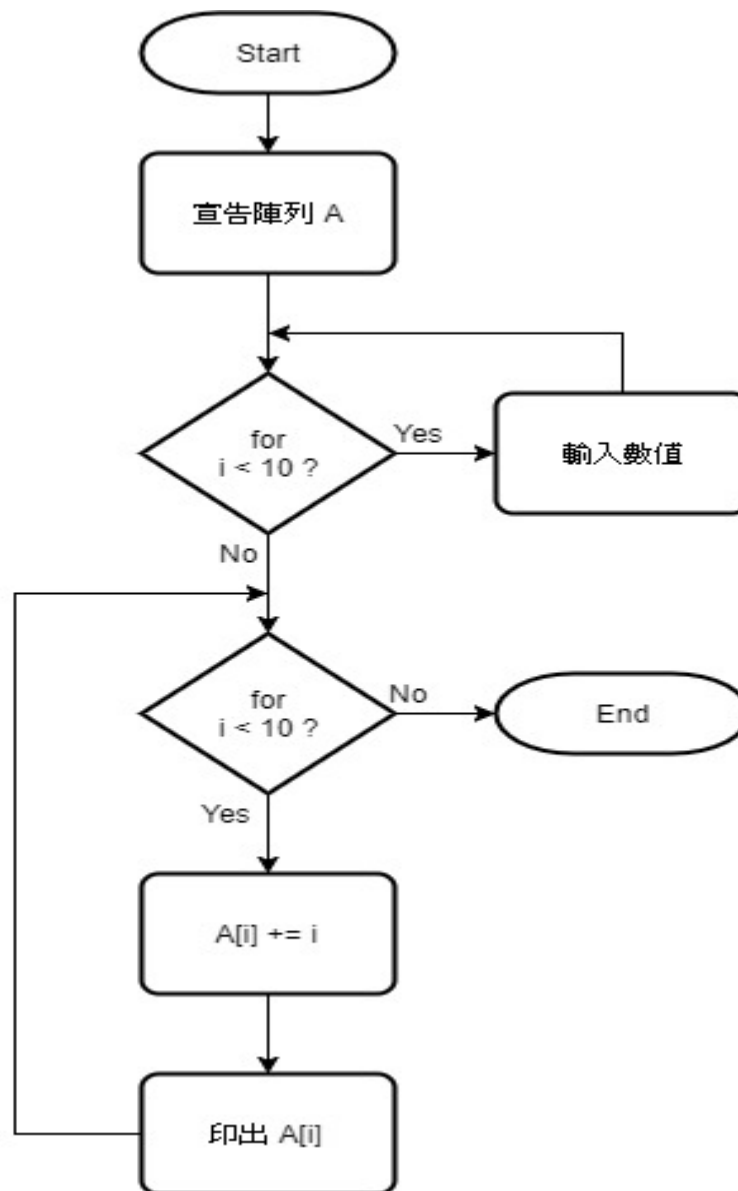
---

### ► 5-1. 參考程式：

```
Sub Main()  
    Dim A(10) As Integer  
    For i = 1 To 10  
        Console.Write("Please input the value: ")  
        A(i) = Console.ReadLine()  
    Next  
    For i = 1 To 10  
        If A(i) > 5 Then  
            A(i) -= 5  
        Else  
            A(i) += 5  
        End If  
        Console.WriteLine("A(" & i & ") = " & A(i))  
    Next  
    Console.Read()  
End Sub
```

## Ch5. 習題

### ► 5-2 流程圖：



## Ch5. 習題

---

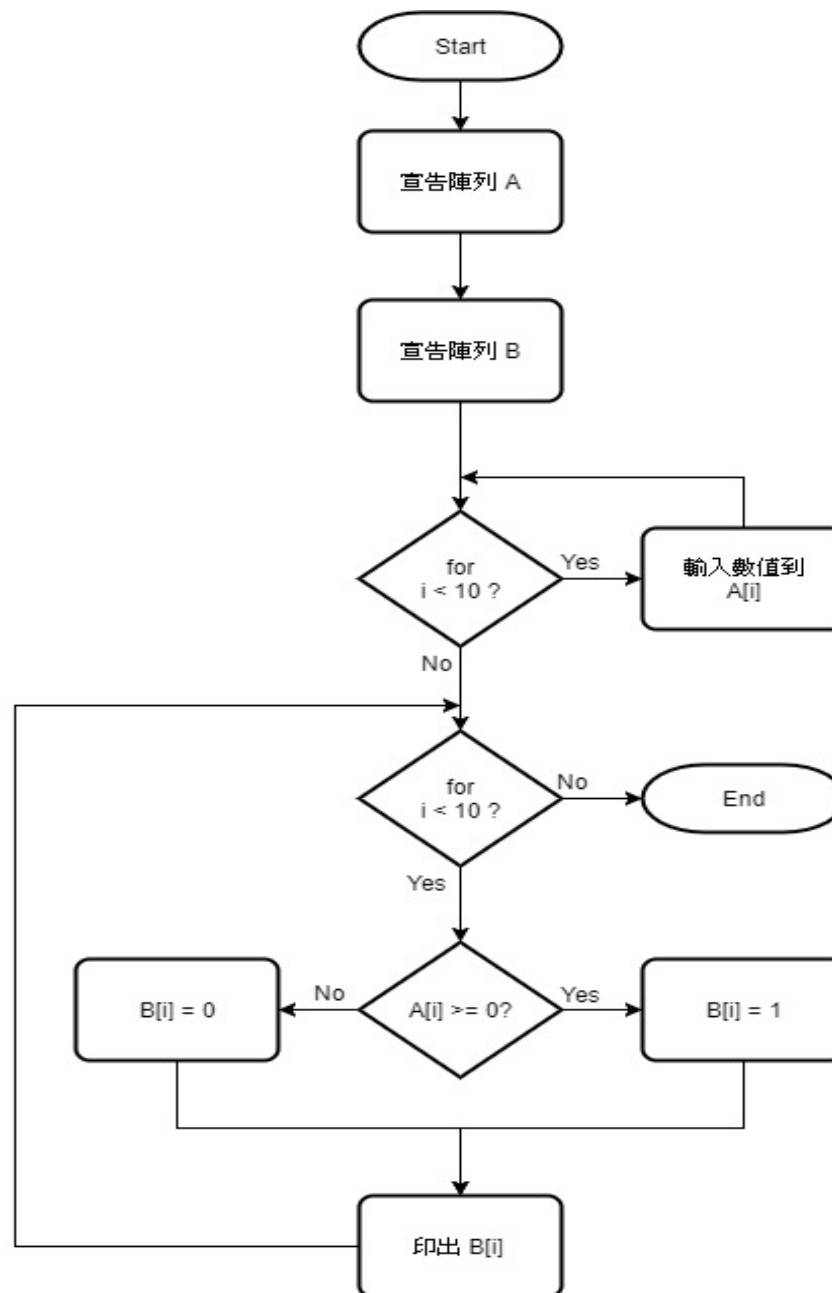
### ► 5-2. 參考程式：

```
Sub Main()  
    Dim A(10) As Integer  
    For i = 1 To 10  
        Console.Write("Please input the value: ")  
        A(i) = Console.ReadLine()  
    Next  
    For i = 1 To 10  
        A(i) += i  
        Console.WriteLine("A(" & i & ") = " & A(i))  
    Next  
    Console.Read()  
End Sub
```



## Ch5. 習題

### ► 5-3 流程圖：



## Ch5. 習題

---

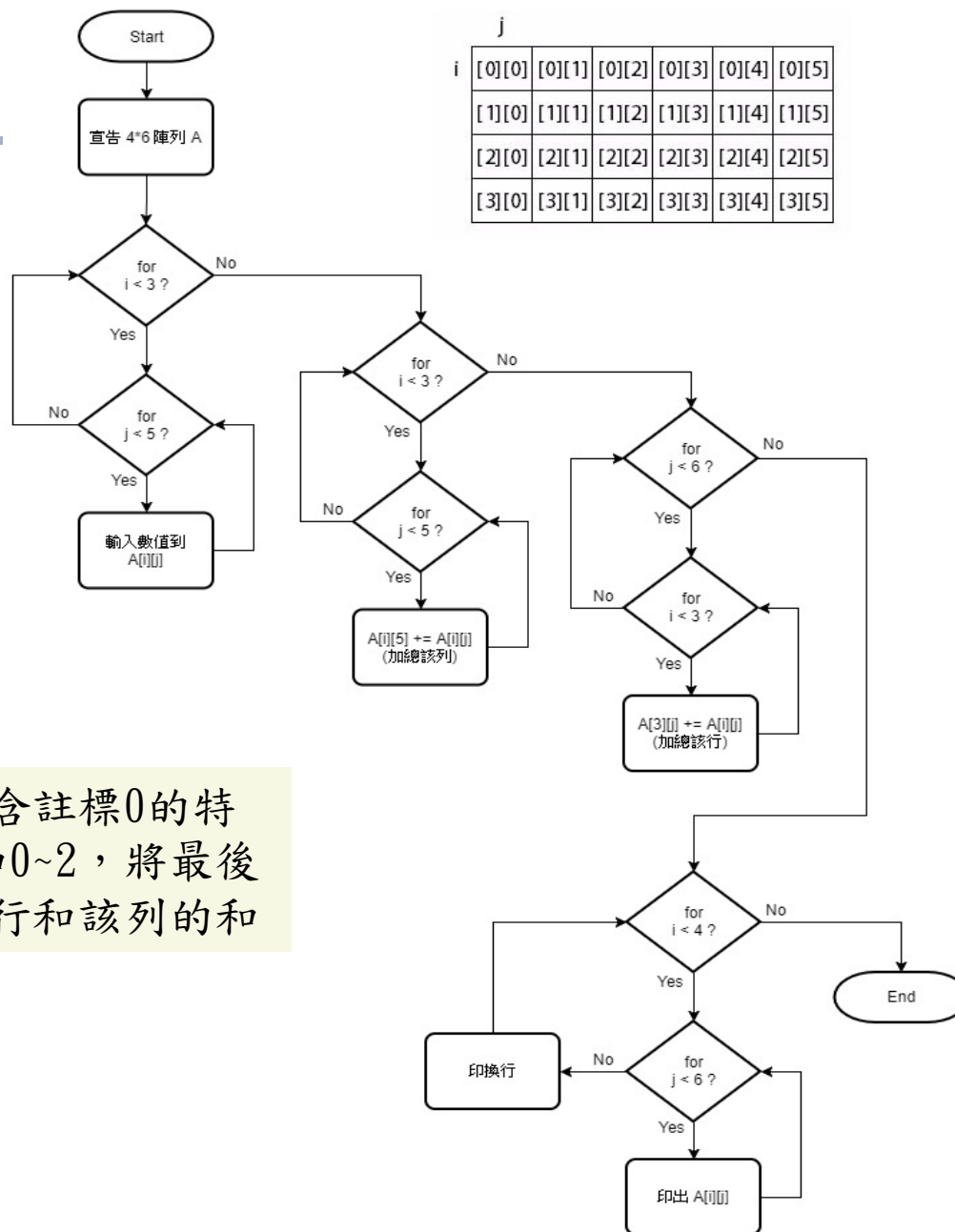
### ► 5-3. 參考程式：

```
Sub Main()  
    Dim A(10), B(10) As Integer  
    For i = 1 To 10  
        Console.Write("Please input the value: ")  
        A(i) = Console.ReadLine()  
    Next  
    For i = 1 To 10  
        If A(i) >= 0 Then  
            B(i) = 1  
        Else  
            B(i) = 0  
        End If  
        Console.WriteLine("B(" & i & ") = " & B(i))  
    Next  
    Console.Read()  
End Sub
```



## Ch5. 習題

### ► 5-4 流程圖：



利用VB陣列宣告時包含註標0的特性，將資料放在0~4和0~2，將最後一行和列用來儲存該行和該列的和

# Ch5. 習題

## ► 5-4. 參考程式：

```
Sub Main()  
    Dim A(3, 5) As Integer  
    For i = 0 To 2 '請求輸入  
        For j = 0 To 4  
            Console.Write("Please input A(" & i & "," & j & "): ")  
            A(i, j) = Console.ReadLine()  
        Next  
    Next  
    For i = 0 To 2 '計算列，將列的和存入每列最後一個元素  
        For j = 0 To 4  
            A(i, 5) += A(i, j)  
        Next  
    Next  
    For j = 0 To 5 '計算行，將行的和存入每列最後一個元素  
        For i = 0 To 2  
            A(3, j) += A(i, j)  
        Next  
    Next  
    For i = 0 To 3 '印出矩陣和結果  
        For j = 0 To 5  
            Console.Write(A(i, j) & vbTab)  
        Next  
        Console.WriteLine()  
    Next  
    Console.Read()  
End Sub
```

## Ch5. 習題

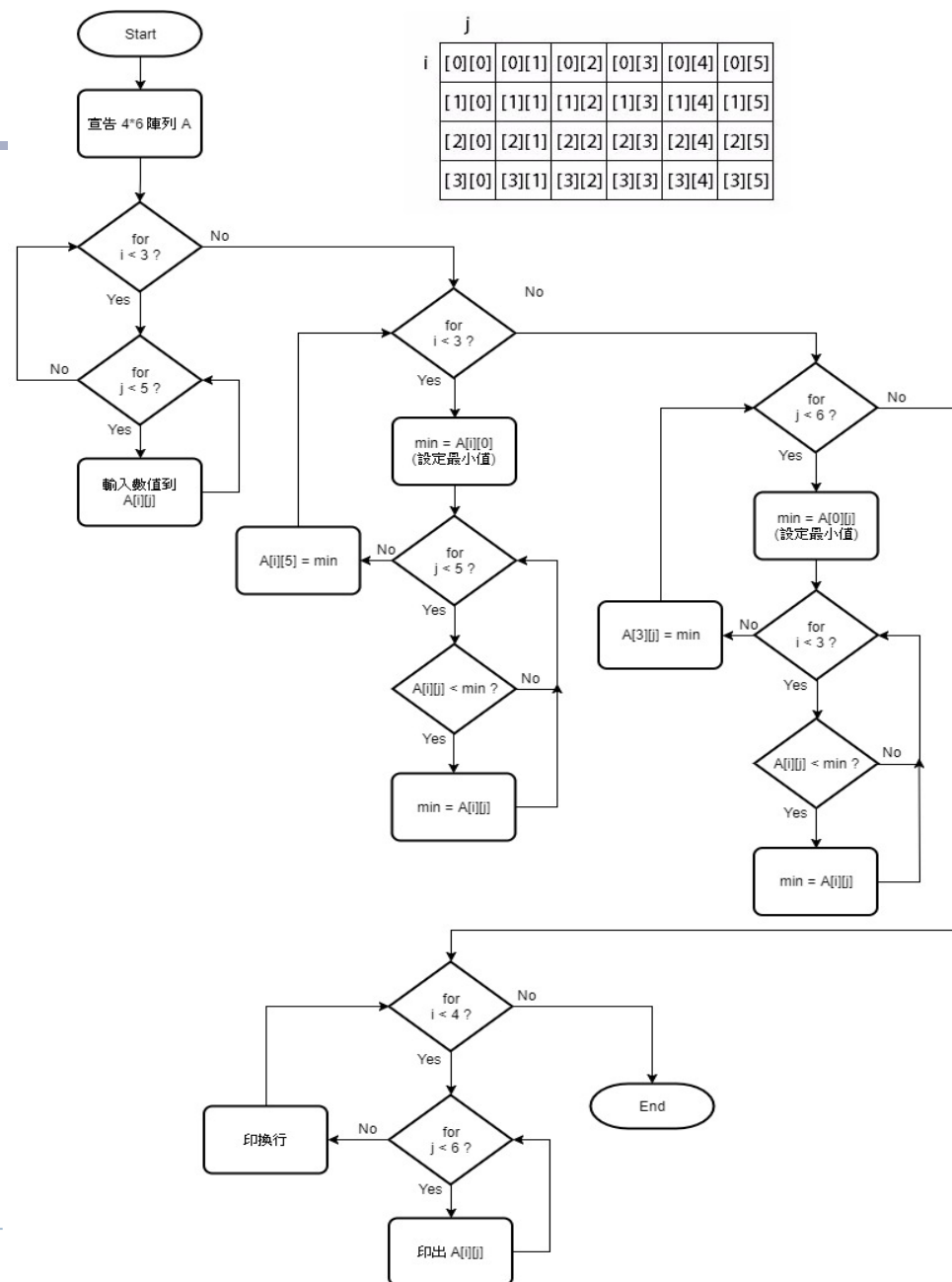
---

- ▶ 5-5. 寫一程式，將15數字存入 $3 \times 5$ 的二維陣列A中，求每一行及每一列數字的最小值。
- ▶ 5-6. 寫一程式，輸入兩組數字： $a_1, a_2, \dots, a_5$ 和 $b_1, b_2, \dots, b_5$ 。求 $a_i + b_i$ ， $i=1$ 到 $i=5$ 。
- ▶ 5-7. 寫一程式，輸入兩組數字： $a_1, a_2, \dots, a_5$ 和 $b_1, b_2, \dots, b_5$ 。令 $x$ 為 $a$ 中的最大值，令 $y$ 為 $b$ 中的最大值，求 $x$ 與 $y$ 中較小者。



# Ch5. 習題

## ► 5-5 流程圖：



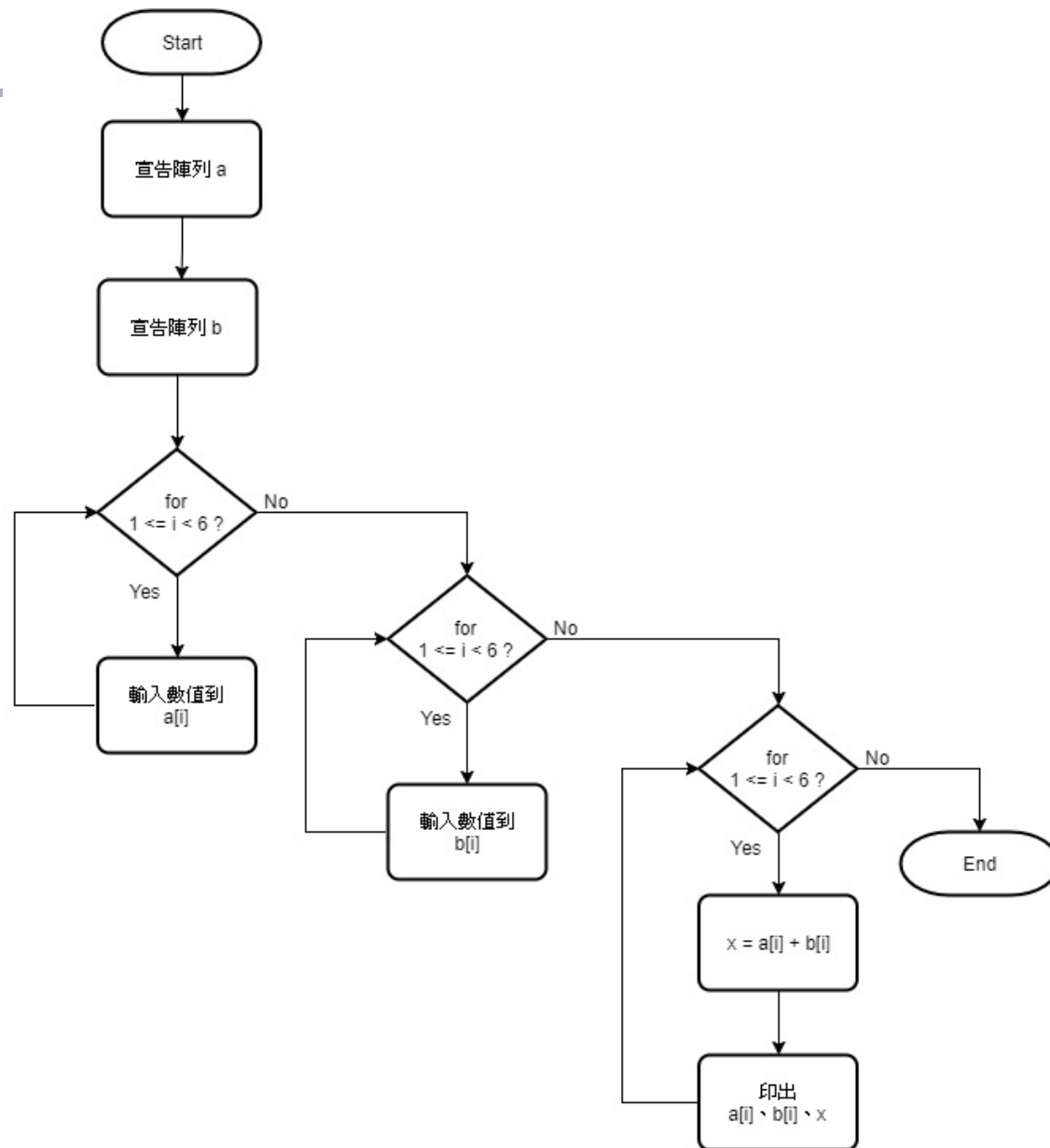
## Ch5. 習題

### ► 5-5. 參考程式：

```
Sub Main()  
    Dim A(3, 5), min As Integer  
    For i = 0 To 2 '請求輸入  
        For j = 0 To 4  
            Console.Write("Please input A(" & i & "," & j & "): ")  
            A(i, j) = Console.ReadLine()  
        Next  
    Next  
    For i = 0 To 2 '尋找列，將列的最小值存入每列最後一個元素  
        min = A(i, 0)  
        For j = 1 To 4  
            If A(i, j) < min Then min = A(i, j)  
        Next  
        A(i, 5) = min  
    Next  
    For j = 0 To 5 '計算行，將行的和存入每列最後一個元素  
        min = A(0, j)  
        For i = 1 To 2  
            If A(i, j) < min Then min = A(i, j)  
        Next  
        A(3, j) = min  
    Next  
    For i = 0 To 3 '印出矩陣和結果  
        For j = 0 To 5  
            Console.Write(A(i, j) & vbTab)  
        Next  
        Console.WriteLine()  
    Next  
    Console.Read()  
End Sub
```

## Ch5. 習題

### ► 5-6 流程圖：



## Ch5. 習題

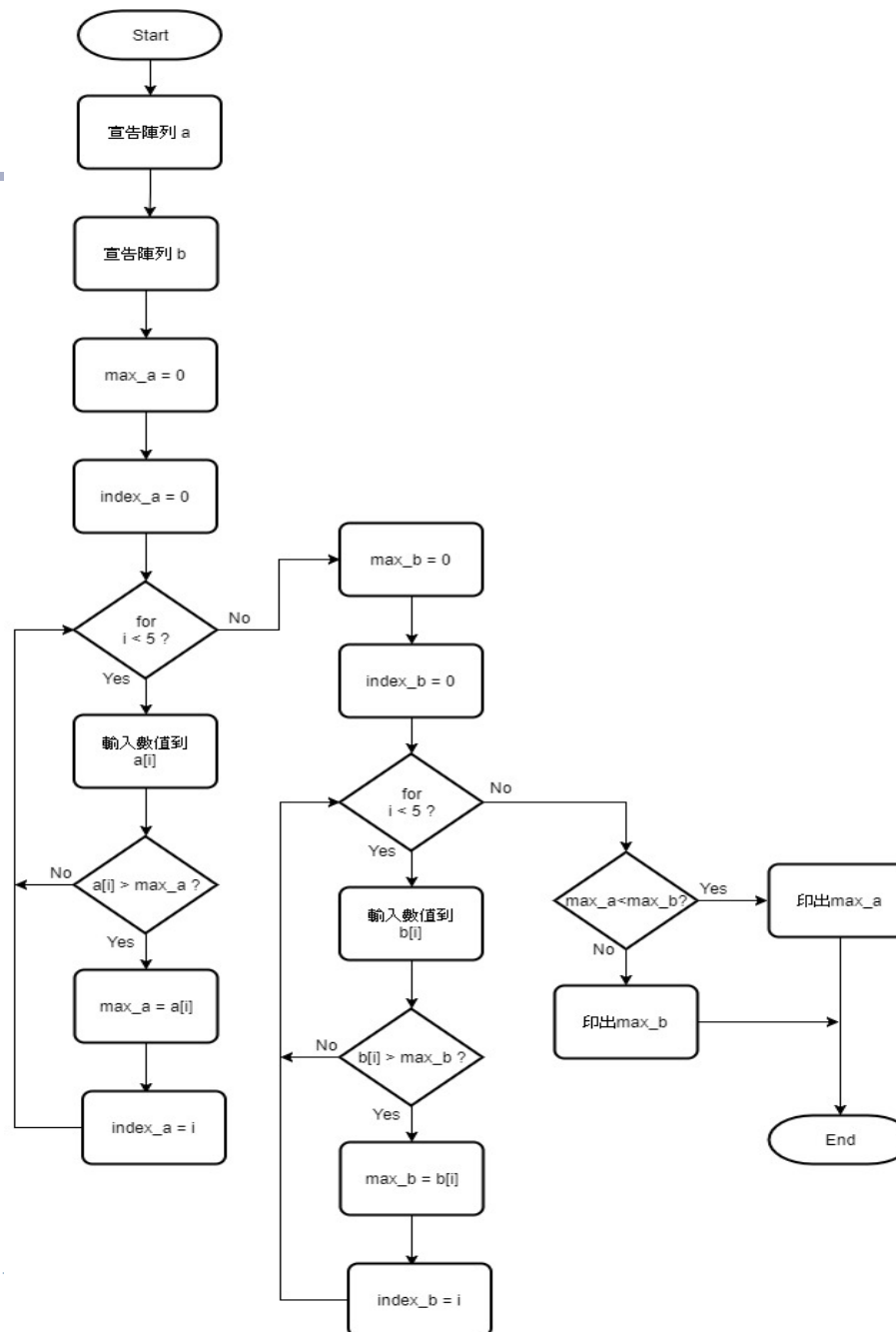
---

### ► 5-6. 參考程式：

```
Sub Main()  
    Dim a(5), b(5) As Integer  
    For i = 1 To 5 '輸入a陣列  
        Console.Write("Please input a(" & i & "): ")  
        a(i) = Console.ReadLine()  
    Next  
    For i = 1 To 5 '輸入b陣列  
        Console.Write("Please input b(" & i & "): ")  
        b(i) = Console.ReadLine()  
    Next  
    For i = 1 To 5 '計算a(i)+b(i)  
        Console.WriteLine("a(" & i & ") + b(" & i &  
            ") = " & a(i) + b(i))  
    Next  
    Console.Read()  
End Sub
```

## Ch5. 習題

### ► 5-7 流程圖：





## Ch5. 習題

### ► 5-7. 參考程式：

```
Sub Main()  
    Dim a(5), b(5), max_a, max_b, index_a, index_b As Integer  
    max_a = 0 : index_a = 0  
    For i = 1 To 5 '輸入a陣列並記錄最大值  
        Console.Write("Please input a(" & i & "): ")  
        a(i) = Console.ReadLine()  
        If a(i) > max_a Then  
            max_a = a(i) : index_a = i  
        End If  
    Next  
    max_b = 0 : index_b = 0  
    For i = 1 To 5 '輸入b陣列並記錄最大值  
        Console.Write("Please input b(" & i & "): ")  
        b(i) = Console.ReadLine()  
        If b(i) > max_b Then  
            max_b = b(i) : index_b = i  
        End If  
    Next  
    If max_a < max_b Then '顯示較小者  
        Console.WriteLine("a(" & index_a & ") = " & max_a)  
    Else  
        Console.WriteLine("b(" & index_b & ") = " & max_b)  
    End If  
    Console.Read()  
End Sub
```

休息一下~

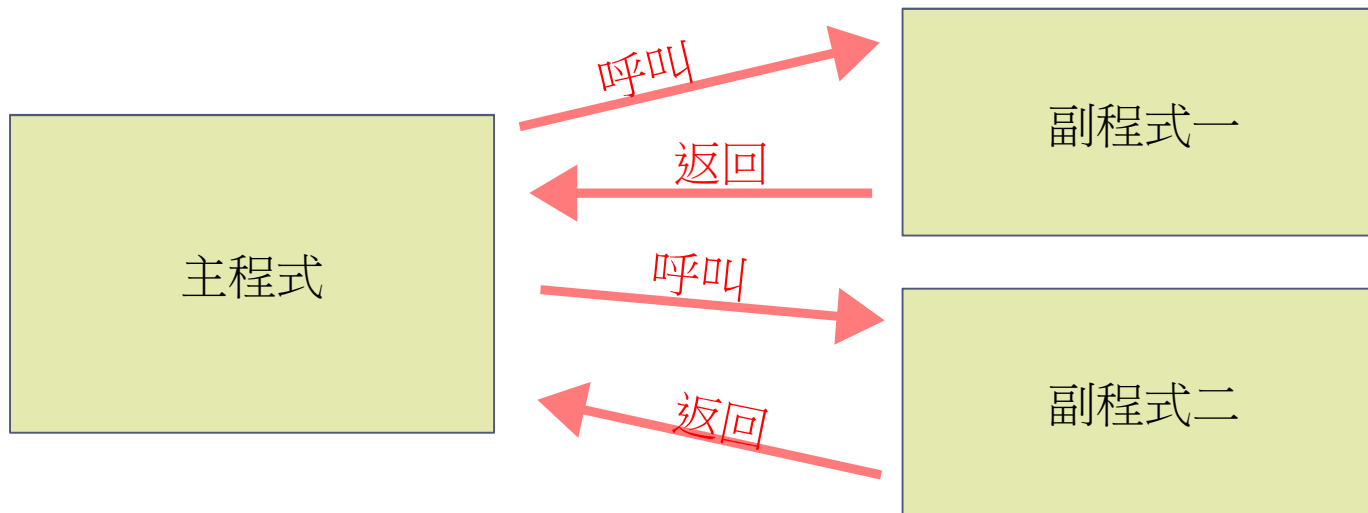
---



## Ch6. 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



## Ch6.副程式

---

- ▶ 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

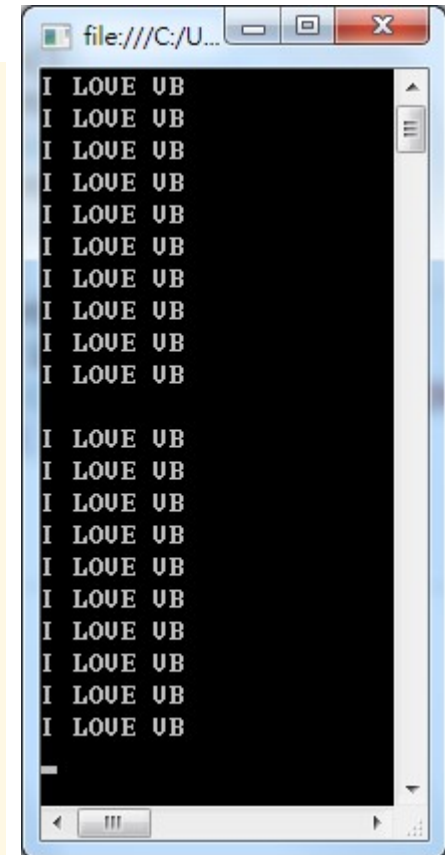
```
Dim A(10) As Integer
Sub Main()
    For i = 1 To 10 '亂數產生數值
        A(i) = Int(Rnd() * 10) + 1
    Next
    Call PrintRnd() '印出陣列內容
    For i = 1 To 10 '個元素*10
        A(i) *= 10
    Next
    PrintRnd() '印出陣列內容
    Console.Read()
End Sub
Sub PrintRnd()
    For i = 1 To 10
        Console.Write(A(i)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

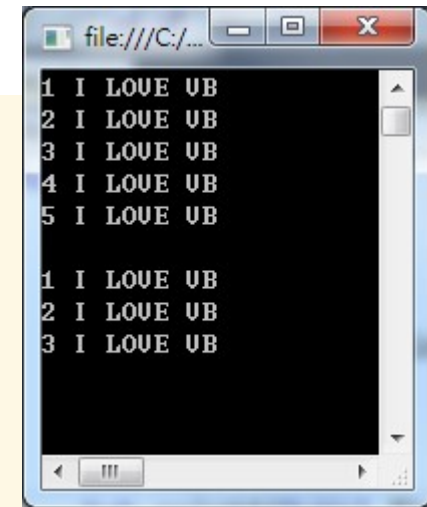


## 副程式(有傳遞參數)

- ▶ 將數值傳給副程式的X：

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

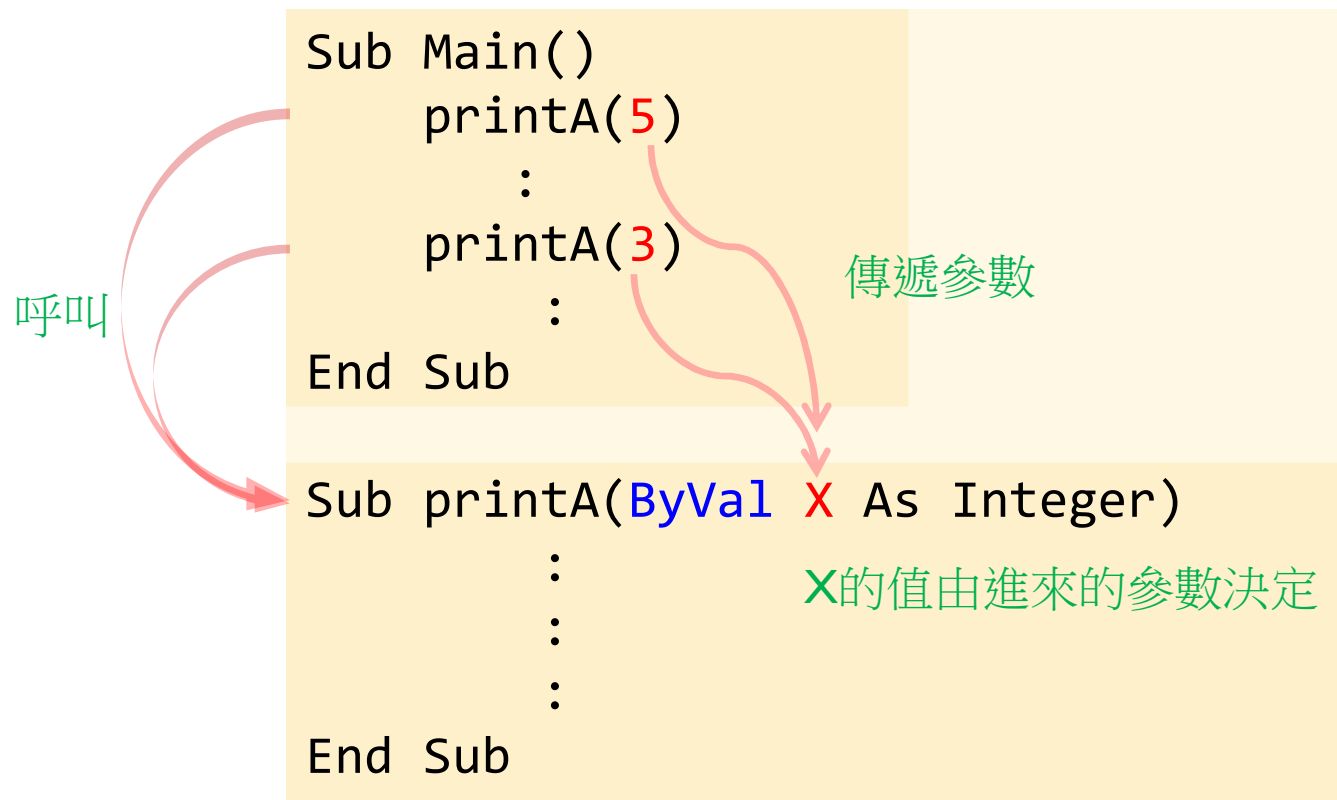
```
Sub printA(ByVal X As Integer)  
    For i = 1 To X  
        Console.WriteLine(i & " I LOVE VB")  
    Next  
End Sub
```





## 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



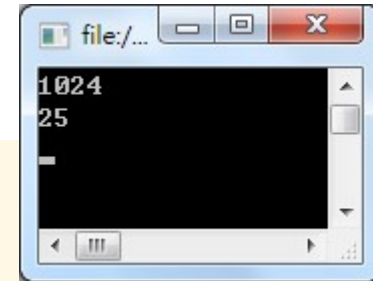
## 副程式(傳遞多個參數)

- ▶ 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
```



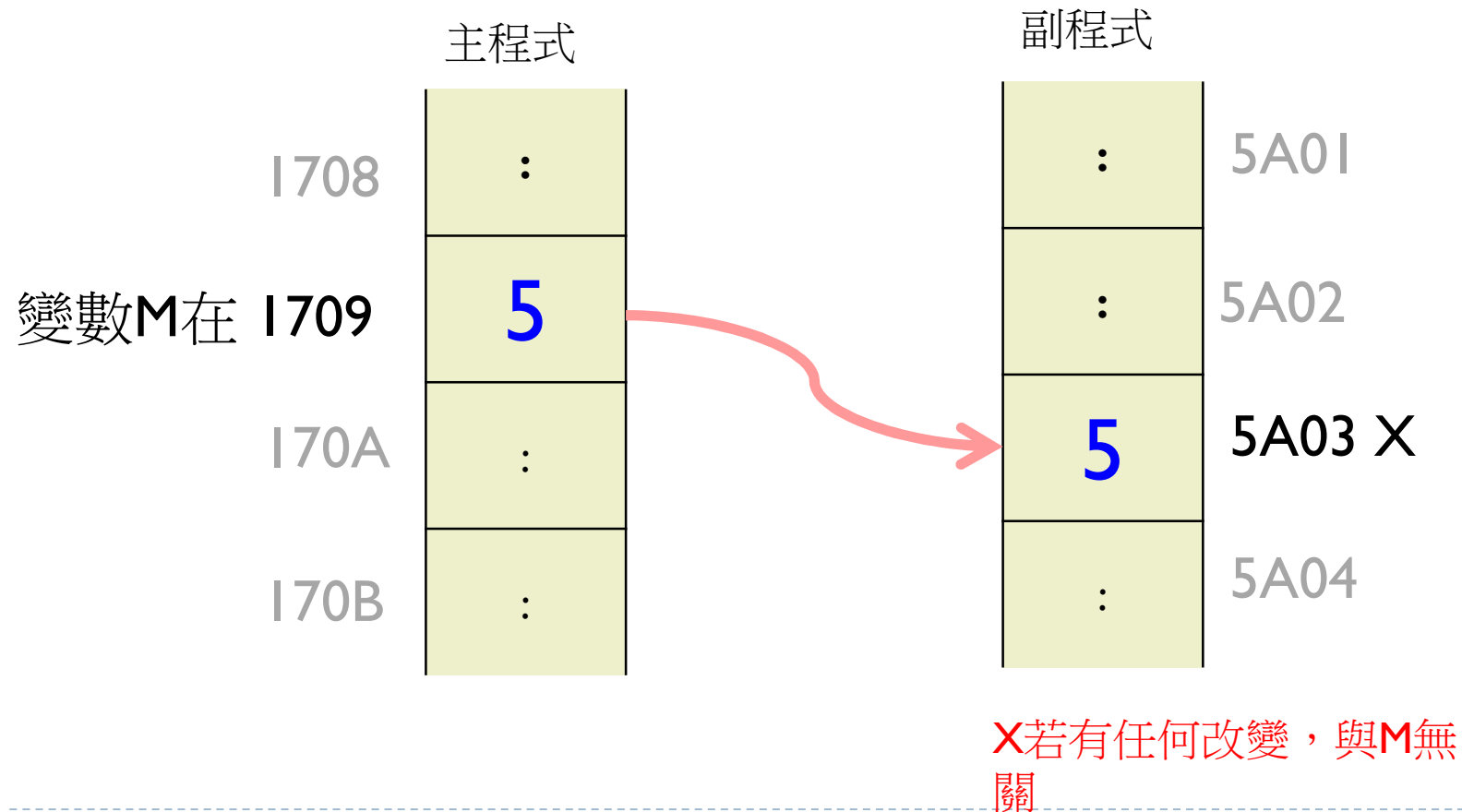
# ByVal 與 ByRef

- ▶ 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

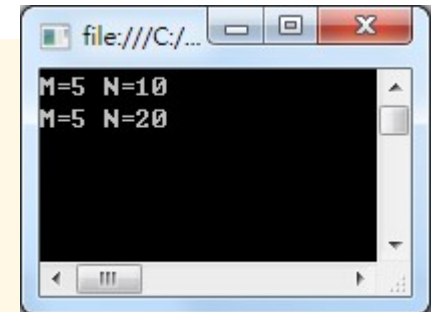


# ByVal 與 ByRef

- ▶ M使用 **ByVal**，N使用 **ByRef**，可以看出N被副程式改變了

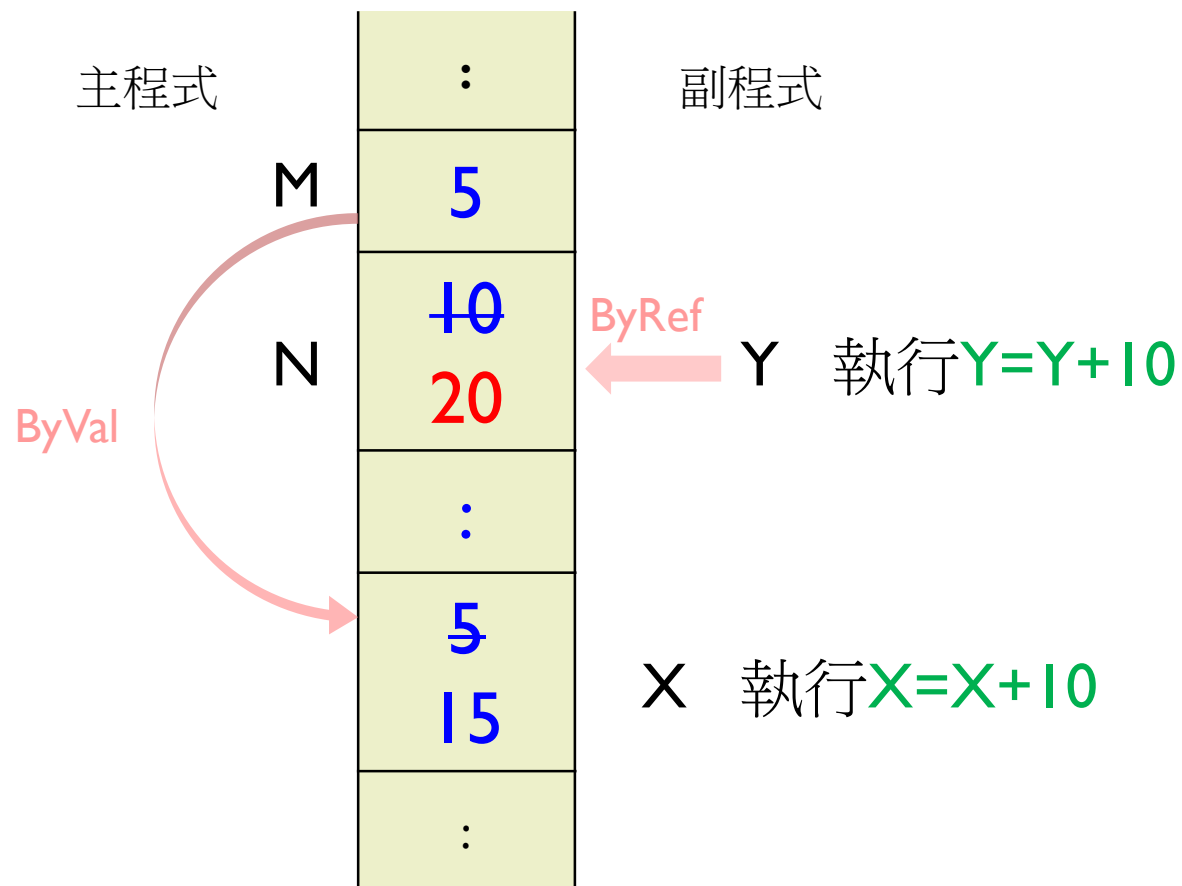
```
Sub Main()  
    Dim M, N As Integer  
    M = 5 : N = 10  
    Console.WriteLine("M=" & M & " N=" & N)  
    printA(M, N)  
    Console.WriteLine("M=" & M & " N=" & N)  
    Console.Read()  
End Sub
```

```
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
```

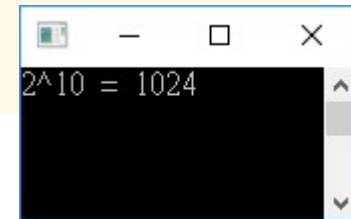
- 與副程式相同，參數可有可無
- 函數中執行到Return敘述則結束函數，並將指定的值傳回

# 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
```

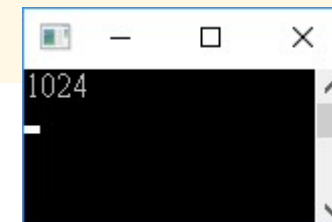


# 函數(Function)

## ▶ 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
```

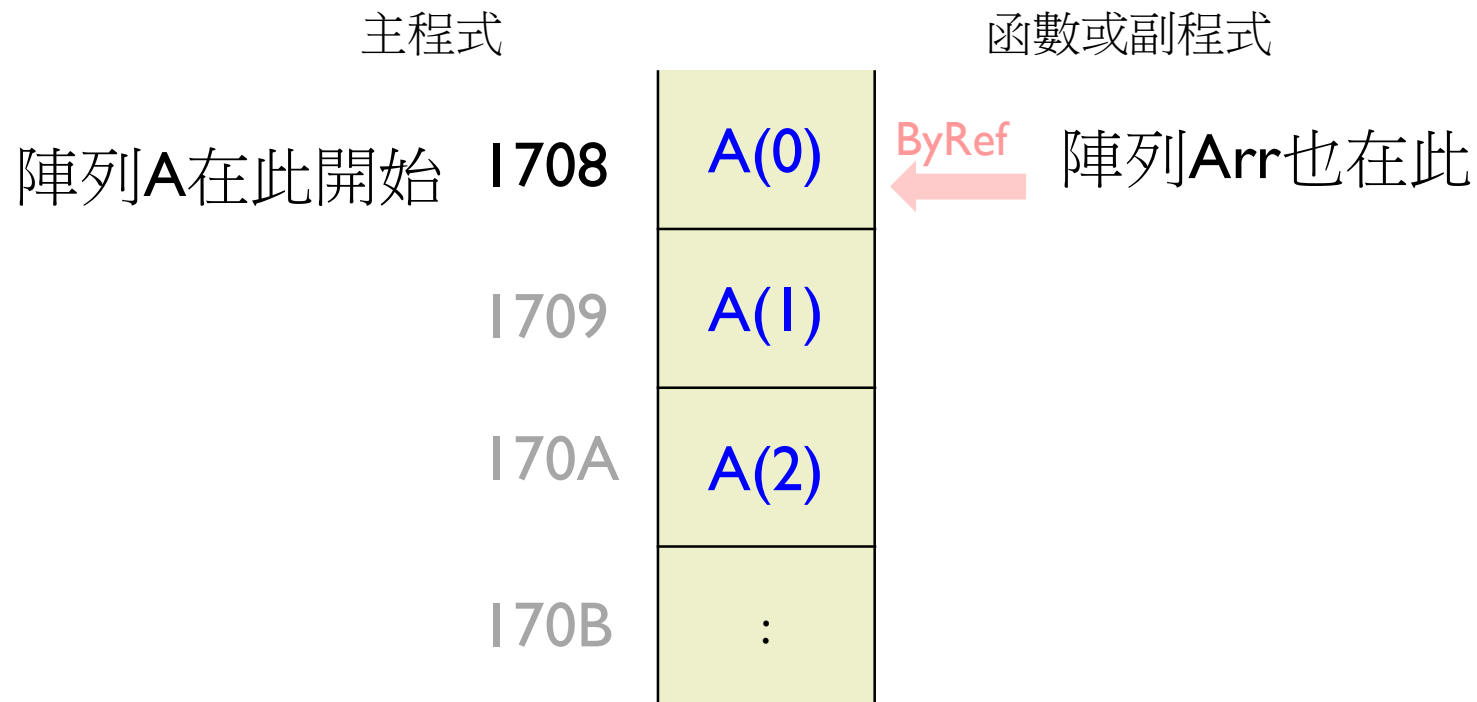
## 傳遞陣列

---

- ▶ 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

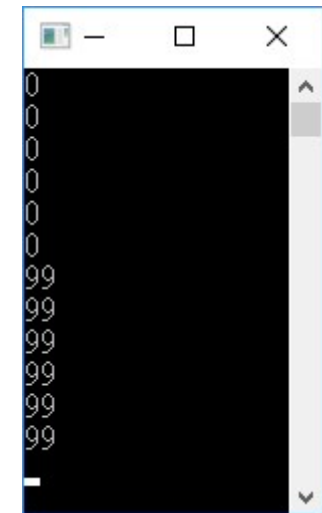
# 傳遞陣列

- ▶ 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

## Ch6. 副程式

---

- ▶ 兩個一維陣列 相加：
  - ▶ 1. 將一組數字讀入A陣列。
  - ▶ 2. 將一組數字讀入B陣列。
  - ▶ 3. 將A陣列與B陣列相加，成為C陣列，列印C陣列。
  
- ▶ 我們需要三個副程式：
  - ▶ 1. 將數字讀入一個陣列的副程式。
  - ▶ 2. 將兩個陣列相加的副程式。
  - ▶ 3. 列印一個陣列的副程式。



## Ch6.副程式

---

- ▶ 1.將數字讀入一個陣列的副程式。

```
Sub input_array(ByRef x() As Integer, ByVal n As Integer)
    Dim i As Integer = 0
    Console.WriteLine("---開始讀取陣列---")
    While i < n
        Console.Write("輸入數字: ")
        x(i) = Console.ReadLine()
        i += 1
    End While
End Sub
```

- ▶ 副程式名稱為input\_array，並需要傳入x及n兩個參數，x是陣列名稱，n是要輸入的數字的數目。
- ▶ 必須在函數名稱後面的參數列作適當的宣告。



## Ch6.副程式

---

- ▶ 2.將兩個陣列相加的副程式。

```
Sub add_array(ByRef x() As Integer, ByRef y() As Integer,  
              ByRef z() As Integer, ByVal n As Integer)  
    Dim i As Integer = 0  
    Console.WriteLine("---開始陣列相加---")  
    While i < n  
        z(i) = x(i) + y(i)  
        i += 1  
    End While  
End Sub
```

- ▶ 副程式名稱為add\_array，並需要傳入x、y、z三個陣列名稱及數字的數目n。



## Ch6.副程式

---

- ▶ 3. 列印一個陣列的副程式。

```
Sub print_array(ByRef x() As Integer, ByVal name As String,  
                ByVal n As Integer)  
    Dim i As Integer = 0  
    Console.WriteLine("---開始印出陣列---")  
    While i < n  
        Console.WriteLine(name & "(" & i & ")" & x(i))  
        i += 1  
    End While  
End Sub
```

- ▶ 副程式名稱為print\_array，並需要傳入陣列名稱x，及要印出的陣列名稱的字母name，及數字的數目n。



## Ch6.副程式

---

### ► 主程式：

```
Sub Main()  
    Dim A(), B(), C() As Integer  
    Dim num As Integer  
    Console.Write("請輸入陣列大小:")  
    num = Console.ReadLine()  
    ReDim A(num) : ReDim B(num) : ReDim C(num)  
    '依序呼叫各個動作的副程式  
    input_array(A, num)  
    input_array(B, num)  
    add_array(A, B, C, num)  
    print_array(A, "A", num)  
    print_array(B, "B", num)  
    print_array(C, "C", num)  
    Console.Read()  
End Sub
```

# Ch6.副程式

## ► 完整程式：

```
Sub input_array(ByRef x() As Integer, ByVal n As Integer)
    Dim i As Integer = 0
    Console.WriteLine("---開始讀取陣列---")
    While i < n
        Console.Write("輸入數字: ")
        x(i) = Console.ReadLine()
        i += 1
    End While
End Sub

Sub add_array(ByRef x() As Integer, ByRef y() As Integer, ByRef z() As Integer,
    ByVal n As Integer)
    Dim i As Integer = 0
    Console.WriteLine("---開始陣列相加---")
    While i < n
        z(i) = x(i) + y(i)
        i += 1
    End While
End Sub

Sub print_array(ByRef x() As Integer, ByVal name As String, ByVal n As Integer)
    Dim i As Integer = 0
    Console.WriteLine("---開始印出陣列---")
    While i < n
        Console.WriteLine(name & "(" & i & ")" & x(i))
        i += 1
    End While
End Sub

Sub Main()
    Dim A(), B(), C() As Integer
    Dim num As Integer
    Console.Write("請輸入陣列大小:")
    num = Console.ReadLine()
    ReDim A(num) : ReDim B(num) : ReDim C(num)
    '依序呼叫各個動作的副程式
    input_array(A, num)
    input_array(B, num)
    add_array(A, B, C, num)
    print_array(A, "A", num)
    print_array(B, "B", num)
    print_array(C, "C", num)
    Console.Read()
End Sub
```

## Ch6.副程式

---

- ▶ 求陣列中最大之數：將5個數字讀入陣列，求陣列中最大之數。我們需要兩個副程式。
  - ▶ 1. 將5個數字讀入一個陣列的副程式。
  - ▶ 2. 求這個陣列中數字最大值的副程式。
- ▶ 使用return敘述將結果回傳給呼叫它的敘述，所以呼叫這個副程式時需準備接收傳回值。
- ▶ VB可以直接傳遞陣列，與傳遞變數相同。
- ▶ UBound( )函數會傳回陣列的長度(最大註標值)





## Ch6.副程式

▶ 參考程式：

```
Sub createArray(ByRef x() As Integer) '這是副程式
    For i = 1 To 5
        Console.Write("Input number:")
        x(i) = Console.ReadLine()
    Next
End Sub

Function getMax(ByRef x() As Integer) '這是函數
    Dim max As Integer = 0
    For i = 1 To UBound(x)
        If x(i) > max Then max = x(i)
    Next
    Return max
End Function

Sub Main()
    Dim A(5) As Integer
    createArray(A) '輸入陣列數值
    Console.WriteLine(getMax(A)) '印出陣列最大值
    Console.Read()
End Sub
```

## Ch6.副程式

---

### ▶ Array + 1 :

- ▶ 1. 讀入5個數字，將這5個數字放入一個陣列中。
- ▶ 2. 對每一個陣列中的數字都加1，然後列印出來。

### ▶ 同前例，使用三個副程式：

- ▶ 1. 將數字讀入一個陣列的副程式。
- ▶ 2. 將兩個陣列的每個元素加1的副程式。
- ▶ 3. 列印一個陣列的副程式。



## Ch6.副程式

### ▶ 參考程式：

```
Sub createArray(ByRef x() As Integer)
    For i = 1 To 5
        Console.Write("Input number:")
        x(i) = Console.ReadLine()
    Next
End Sub

Sub addition(ByRef x() As Integer)
    Dim max As Integer = 0
    For i = 1 To UBound(x)
        x(i) += 1
    Next
End Sub

Sub printArray(ByRef x() As Integer)
    Console.WriteLine("After addition:")
    For i = 1 To 5
        Console.WriteLine(x(i))
    Next
End Sub

Sub Main()
    Dim A(5) As Integer
    createArray(A) '輸入陣列數值
    addition(A)    '將每個陣列元素+1
    printArray(A)  '印出陣列
    Console.Read()
End Sub
```

## Ch6. 副程式

---

- ▶ 求小於 $N$ 的所有質數。
  - ▶ 質數 (Prime number)，又稱素數，指在大於1的自然數中，除了1和該數自身外，無法被其他自然數整除的數
  - ▶ 例如假設 $N=20$ ，則小於20的質數有1, 2, 3, 5, 7, 11, 13, 17, 19。
- ▶ 我們需要一個副程式，這個副程式輸入一個正整數 $x$ ，然後判斷 $x$ 是否是一個質數。



## Ch6. 副程式

### ► 流程圖：

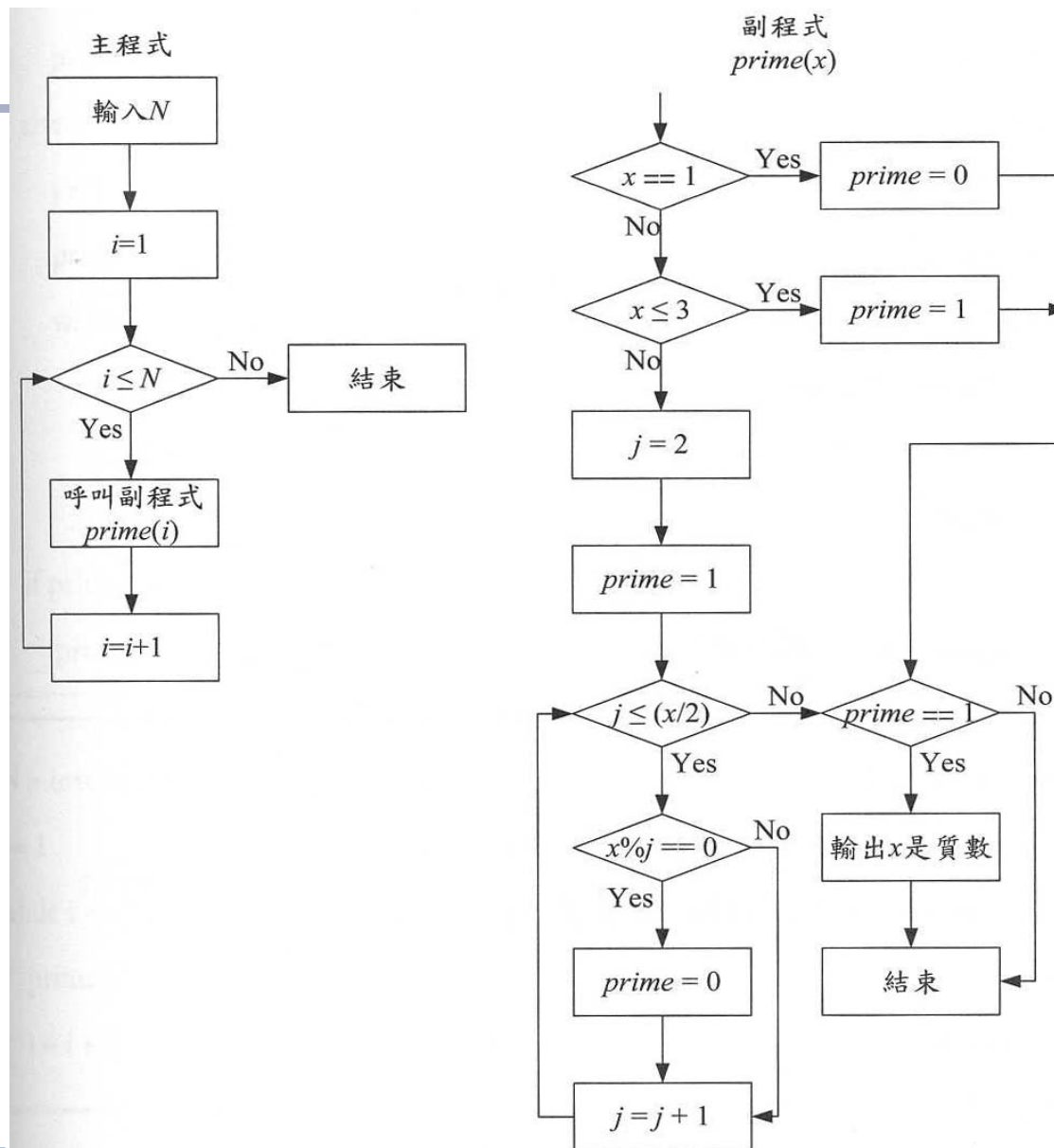


圖 6.1 ► 例題 6.4 的流程圖(決定  $x$  是否為質數)

## Ch6.副程式

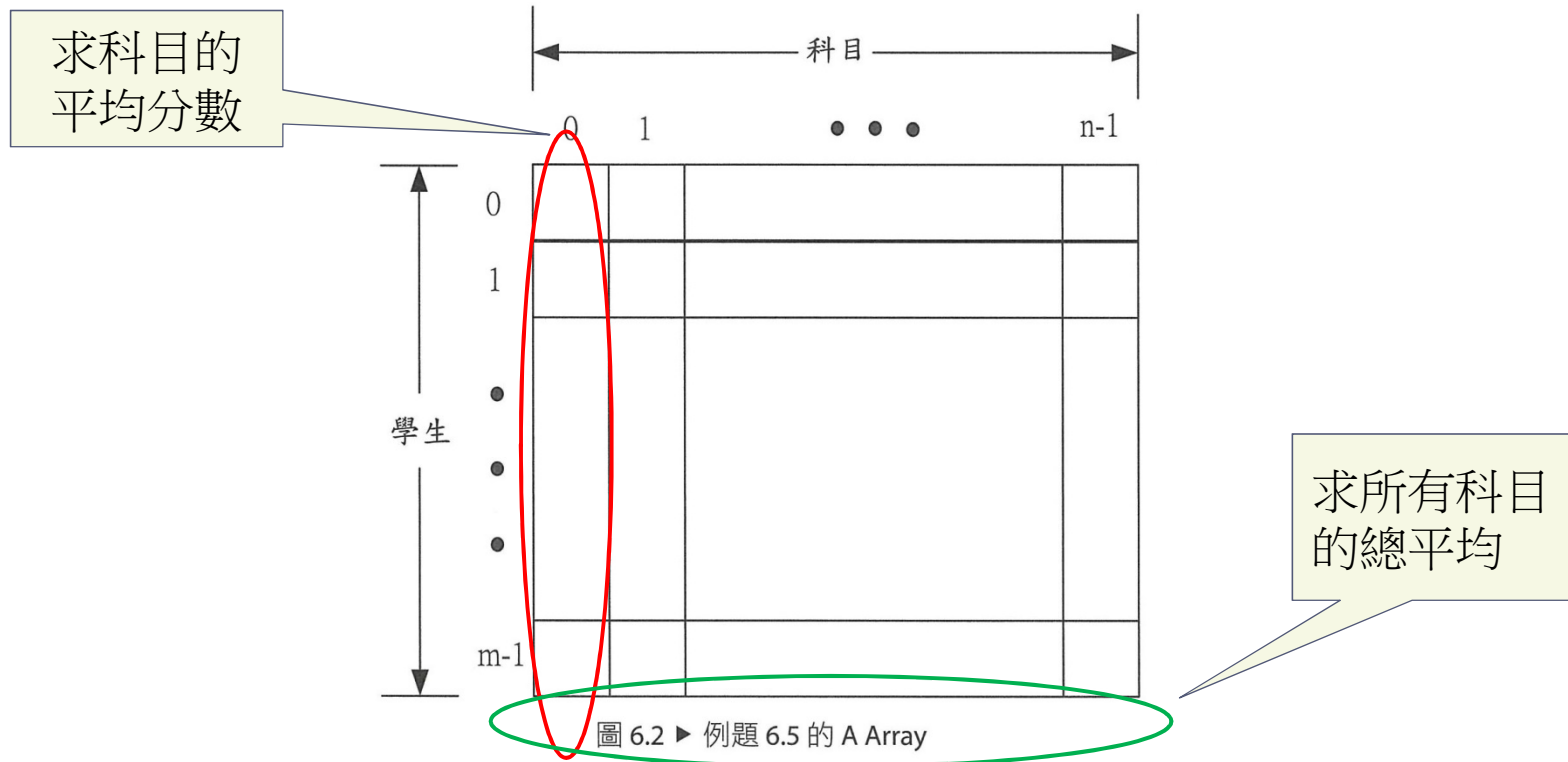
### ▶ 參考程式：

```
Sub prime(ByVal x As Integer)
    Dim prime, j As Integer
    If x = 1 Then '1不列入質數
        prime = 0
    ElseIf x <= 3 Then
        prime = 1 '2、3是質數，不用計算
    Else
        j = 2 : prime = 1
        While j <= (x / 2) '算到x的1/2即可，再加上去也不可能整除x了
            If x Mod j = 0 Then prime = 0
            j += 1
        End While
    End If
    If prime = 1 Then Console.WriteLine(x & " 是一個質數")
End Sub

Sub Main()
    Dim N, i As Integer
    Console.Write("請輸入一個正整數 N: ")
    N = Console.ReadLine() '不能輸入0或1，會沒有輸出訊息
    i = 1
    While i <= N
        prime(i)
        i += 1
    End While
    Console.Read()
End Sub
```

## Ch6. 副程式

- ▶ 求出各科平均分數及總平均分數。
  - ▶ 我們有 $m$ 個學生，每一個學生有 $n$ 門課，我們的任務是對每一門課，求這門課的平均分數，然後求所有科目的總平均。



# Ch6.副程式

---

▶ 參考程式：

```
'輸入m個學生的n科成績
Sub input2DArray(ByVal m As Integer, ByVal n As Integer, ByRef A(,) As Single)
    For i = 1 To m
        Console.WriteLine("--請輸入第" & i & "位學生的成績--")
        For j = 1 To n
            Console.Write("輸入第" & j & "科:")
            A(i, j) = Console.ReadLine()
        Next
    Next
End Sub

'求第i科平均成績
Function subjectAverage(ByVal i As Integer, ByVal m As Integer,
                        ByRef A(,) As Single)
    Dim sum As Single = 0
    For j = 1 To m
        sum += A(j, i)
    Next
    sum /= m
    Console.WriteLine("第" & i & "科平均為:" & sum)
    Return sum
End Function
```



# Ch6.副程式

---

▶ 參考程式：

```
'求各科總平均
Sub getAverage(ByVal m As Integer, ByVal n As Integer,
    ByVal A(,) As Single)
    Dim sum As Single = 0
    For i = 1 To n
        sum += subjectAverage(i, m, A)
    Next
    sum /= n
    Console.WriteLine("總平均:" & sum)
End Sub
'主程式
Sub Main()
    Dim A(,) As Single
    Dim m, n As Integer
    Console.Write("請輸入學生數目:") : m = Console.ReadLine()
    Console.Write("請輸入科目數:") : n = Console.ReadLine()
    ReDim A(m, n)
    input2DArray(m, n, A)
    getAverage(m, n, A)
    Console.Read()
End Sub
```

## Ch6.副程式

---

- ▶ 判斷日期先後順序。
  - ▶ 輸入日期A與日期B，若：
  - ▶ 日期A在日期B之前，輸出「日期A在日期B之前」。
  - ▶ 日期A在日期B之後，輸出「日期A在日期B之後」。
  - ▶ 日期A在日期B相同，輸出「日期A在日期B相同」。
- ▶ 假設輸入格式為(日/月/年)「21/01/2018」存入變數temp，則使用split()函數可將字串拆開。

```
temp = split("/")
```

- ▶ 以「/」為分隔，拆成三組字串，存入temp陣列，使得temp(0)="21"，temp(1)="01"，temp(2)="2018"



## Ch6. 副程式

### ► 流程圖：

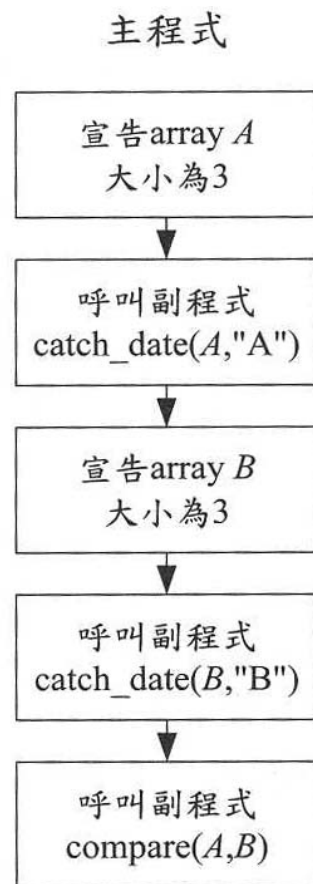


圖 6.3 ► 例題 6.6 主程式的流程圖

### 副程式 *catch\_date(X, name)*

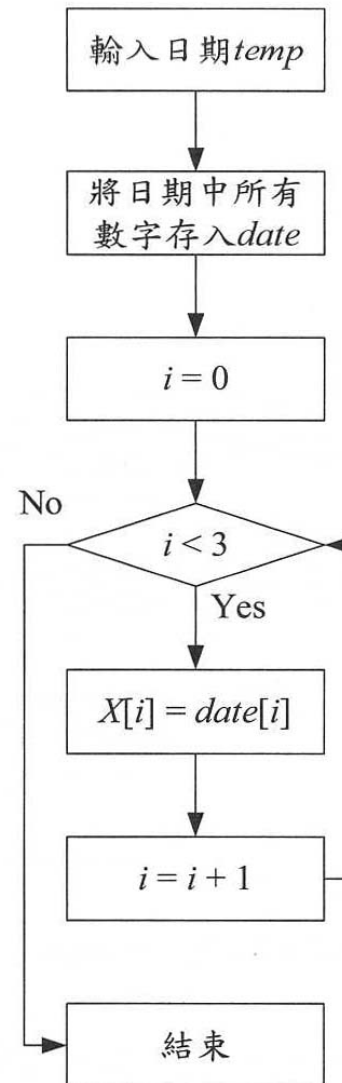


圖 6.4 ► 例題 6.6 副程式 catch-date 的流程圖

## Ch6. 副程式

### ► 流程圖：

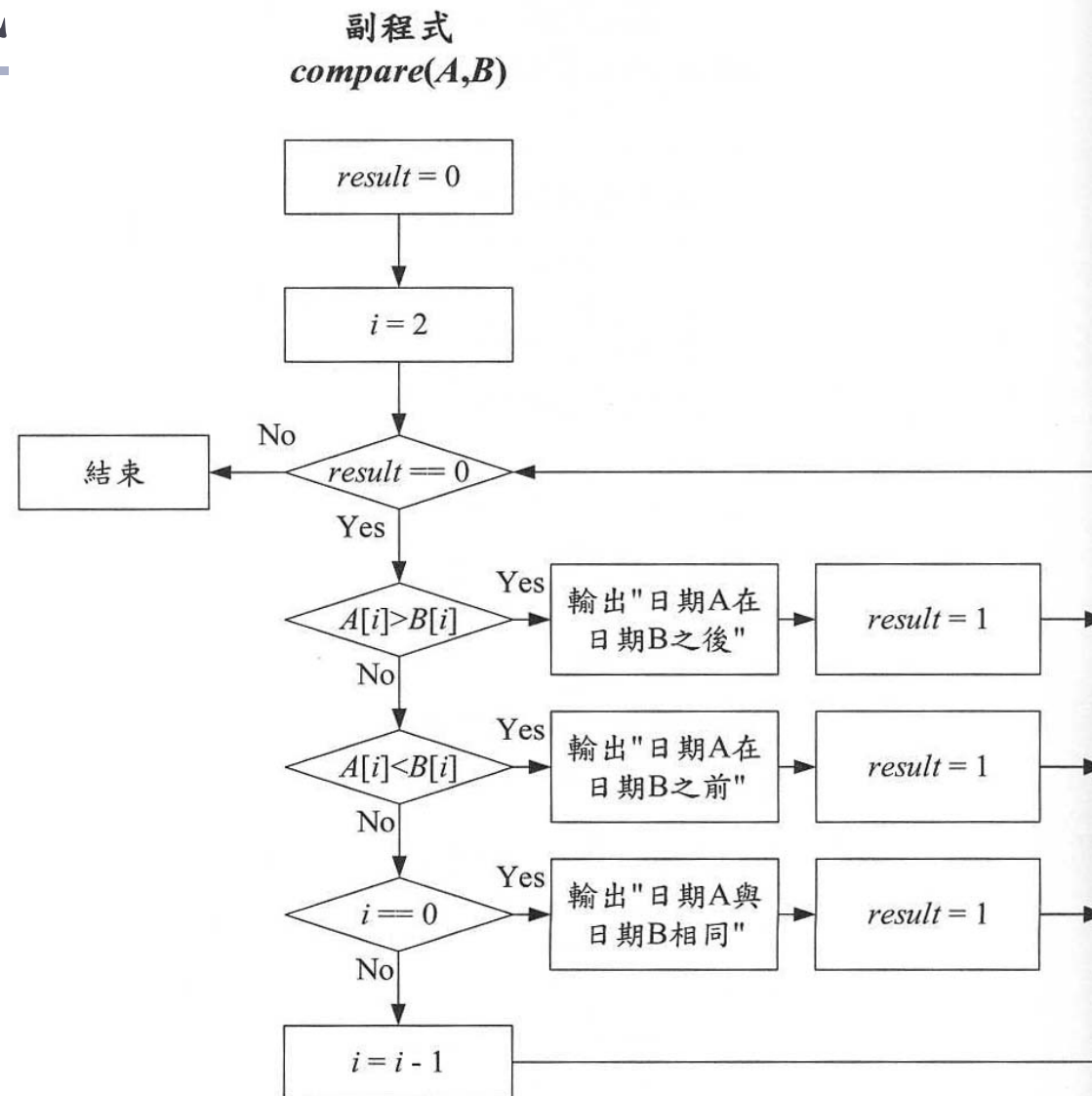


圖 6.5 ► 例題 6.6 副程式 *compare* 的流程圖

## Ch6. 副程式

▶ 參考程式：

‘比較兩日期副程式

```
Sub compare(ByVal A() As String, ByVal B() As String)
    Dim result, I As Integer
    result = 0
    I = 2
    While result = 0
        If A(I) > B(I) Then
            Console.WriteLine("日期A在日期B之後")
            result = 1
        ElseIf A(I) < B(I) Then
            Console.WriteLine("日期A在日期B之前")
            result = 1
        ElseIf I = 0 Then
            Console.WriteLine("日期A與日期B相同")
            result = 1
        Else
            I -= 1
        End If
    End While
End Sub
```

## Ch6. 副程式

---

▶ 參考程式：

'輸入日期副程式

```
Sub catch_date(ByRef A() As String, ByVal day As Char)
    Dim temp As String
    Console.Write("請輸入日期 " & day & " (日/月/年):")
    temp = Console.ReadLine()
    A = Split(temp, "/")
End Sub
```

'主程式

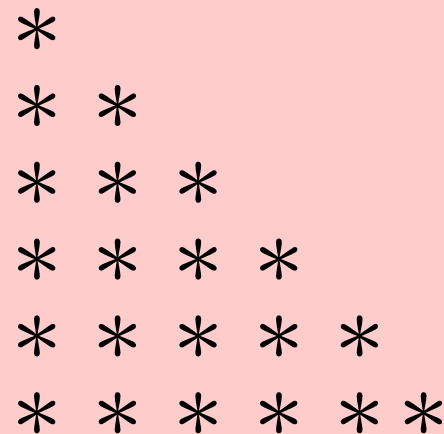
```
Sub Main()
    Dim A(3), B(3) As String
    catch_date(A, "A")
    catch_date(B, "B")
    compare(A, B)
    Console.Read()
End Sub
```



## Ch6. 副程式

---

- ▶ 輸出「\*」三角形，如圖。
- ▶ 印完後要顯示“Enter x to quit, or any other key to continue:”，如果使用者按x鍵，程式就要停止，其他任何鍵就會再印一次星號



```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * * *
```



## Ch6. 副程式

### ► 流程圖：

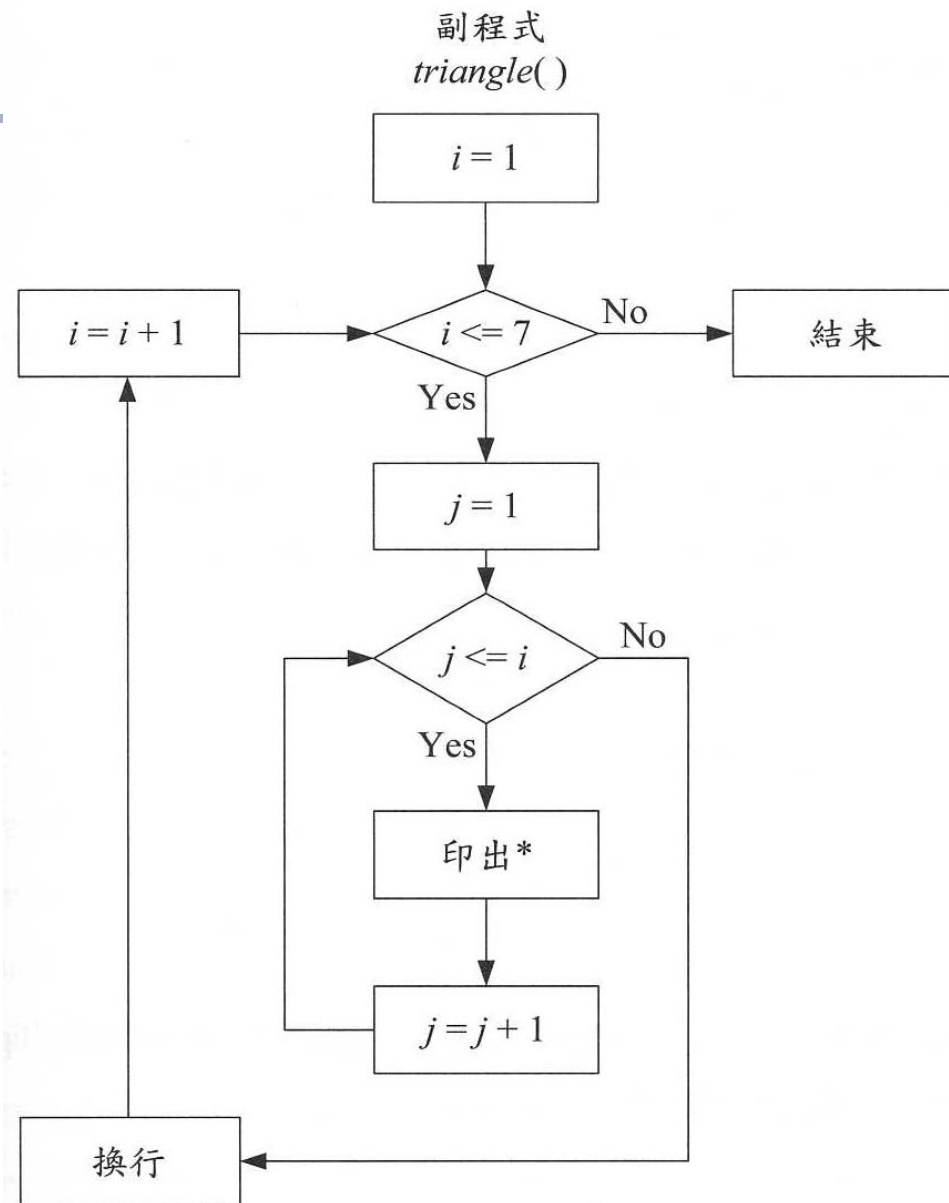


圖 6.6 ► 例題 6.7 副程式的流程圖



## Ch6. 副程式

---

▶ 參考程式：

```
Sub triangle()  
    For i = 1 To 6  
        For j = 1 To i  
            Console.Write("*")  
        Next  
        Console.WriteLine("")  
    Next  
End Sub  
  
Sub Main()  
    Dim ch As Char  
    triangle()  
    Console.Write("Enter x to quit, or any other key to continue:")  
    ch = Console.ReadLine  
    While ch <> "x"  
        triangle()  
        Console.Write("Enter x to quit, or any other key to continue:")  
        ch = Console.ReadLine  
    End While  
End Sub
```

## Ch6.副程式

---

- ▶ 密文解碼。
- ▶ 為了保密，可以將原文加密，加密的方法有很多種，最簡單的英文加密方法是將每一個字母做位移，假設只考慮26個英文大寫字母，且位移規則如下：

原來字母	A	B	C	...	X	Y	Z
轉換後字母	D	E	F	...	A	B	C

小寫字母規則亦同。可以看出皆是往後移三個字母。

- ▶ 還原的動作叫解密，假設加密後的文字為DSSOH和CRR，解密後的文字就是APPLE和ZOO。



## Ch6.副程式

### ► 流程圖：

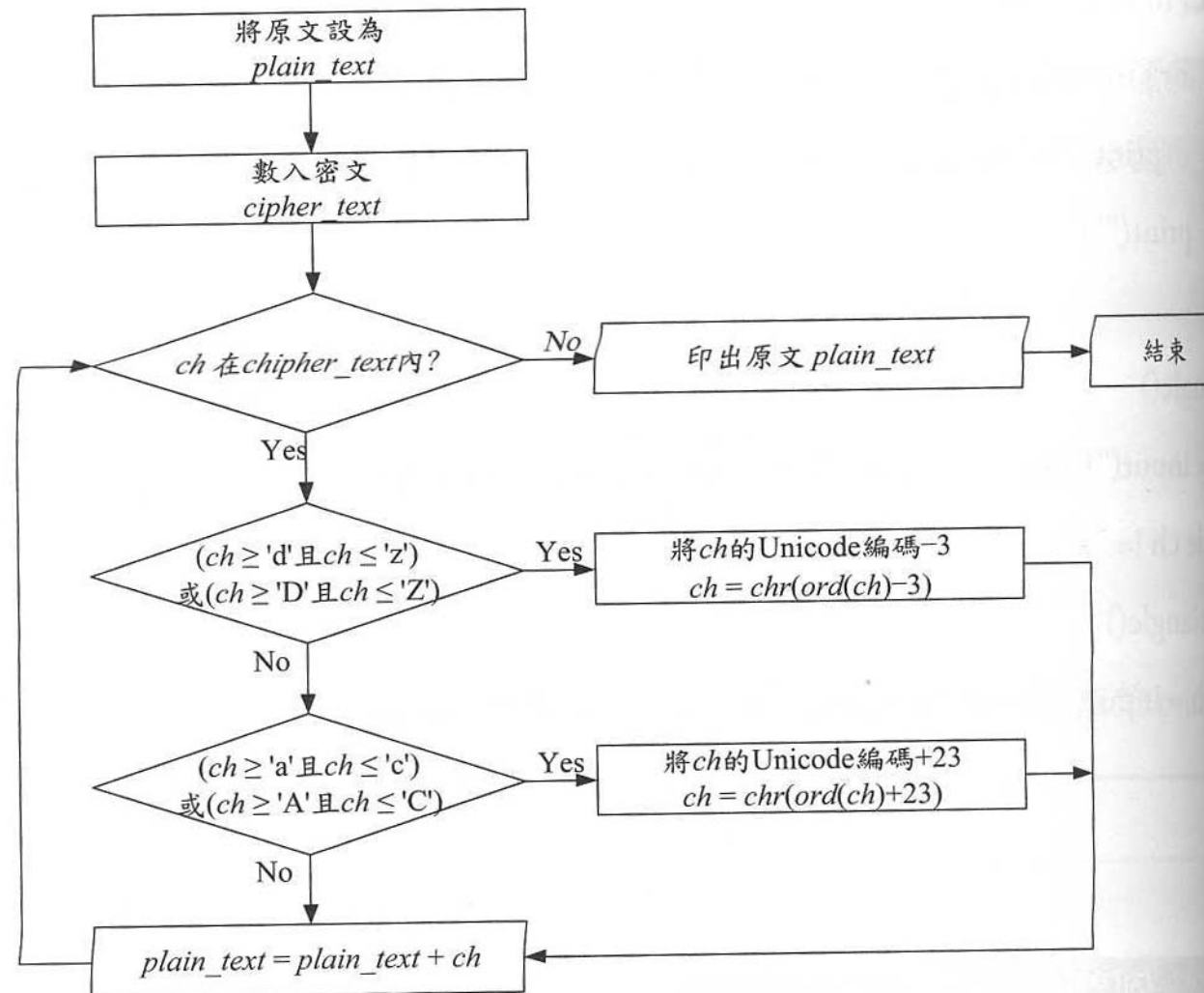


圖 6.7 ► 例題 6.8 的流程圖

## Ch6.副程式

---

### ▶ 參考程式：

```
Sub Main()  
    Dim plain_text, cipher_text As String  
    plain_text = ""  
    Console.Write("請輸入密文:")  
    cipher_text = Console.ReadLine()  
    For Each ch In cipher_text  
        If (ch >= "d" And ch <= "z") Or (ch >= "D" And ch <= "Z") Then  
            ch = Chr(Asc(ch) - 3)  
        ElseIf (ch >= "a" And ch <= "c") Or (ch >= "A" And ch <= "C") Then  
            ch = Chr(Asc(ch) + 23)  
        End If  
        plain_text += ch  
    Next  
    Console.WriteLine("原文為: " & plain_text)  
    Console.Read()  
End Sub
```



## Ch6.副程式

---

- ▶ 說明：
- ▶ 在ASCII碼中，字母A的編碼為 $65_{10}$ ，則函數

`Asc("A")` 會傳回A的編碼65

`Chr(65)` 會傳回字母A

- ▶ 所以字母也可以加或減，會得到另一個字母。
- ▶ 這裡使用了`For Each`指令，它可以將字元一個一個從字串取出來。



## Ch6. 習題

---

- ▶ 6-1. 在主程式中接受使用者輸入梯形之上底、下底與高的值，並呼叫一副程式，將之上底、下底與高的值傳入該副程式後，在副程式中計算並印出梯形面積的值。
- ▶ 6-2. 寫一副程式，接受主程式傳進的陣列與陣列大小，分別計算陣列上第奇數個元素與第偶數個元素之平均值並印出。
- ▶ 6-3. 在主程式中接受使用者輸入A、B、C的值並呼叫一副程式，將A、B、C的值傳入副程式中，在副程式中判斷 $|A|$ 、 $|B|$ 、 $|C|$ 之大小順序，回傳絕對值最大者並在主程式中印出， $|A|$ 、 $|B|$ 、 $|C|$ 之最大值。



## Ch6. 習題

---

### ► 6-1. 參考程式：

```
Function trapezoid(ByVal A As Single, ByVal B As Single,  
                  ByVal H As Single)  
    Return ((A + B) * H) / 2  
End Function  
  
Sub Main()  
    Dim a, b, h As Single  
    Console.Write("請輸入上底:") : a = CSng(Console.ReadLine())  
    Console.Write("請輸入下底:") : b = CSng(Console.ReadLine())  
    Console.Write("請輸入高:") : h = CSng(Console.ReadLine())  
    Console.WriteLine("此梯形面積為:" & trapezoid(a, b, h))  
    Console.Read()  
End Sub
```



## Ch6. 習題

### ► 6-2. 參考程式：

### ► 函數Csng()是將 字串轉成浮點數

```
Sub averageElement(ByRef X() As Single, ByVal L As Integer)
    Dim even, odd As Single : Dim c1, c2 As Integer
    even = 0 : odd = 0 : c1 = 0 : c2 = 0
    For i = 1 To L
        If i Mod 2 = 0 Then
            even += X(i) : c2 += 1
        Else
            odd += X(i) : c1 += 1
        End If
    Next
    Console.WriteLine("奇數元素平均:" & odd / c1)
    Console.WriteLine("偶數元素平均:" & even / c2)
End Sub

Sub Main()
    Dim A() As Single : Dim L As Integer
    Console.Write("請輸入有多少個元素:")
    L = Console.ReadLine()
    ReDim A(L)
    For i = 1 To L
        Console.Write("請輸入元素" & i & ":")
        A(i) = CSng(Console.ReadLine())
    Next
    averageElement(A, L)
    Console.Read()
End Sub
```



## Ch6. 習題

---

### ► 6-3. 參考程式：

```
Function maxABS(a, b, c) '參數a,b,c若未宣告型態，則設定為同傳入的參數型態  
    Dim max As Integer '但不建議，還是仔細宣告好。  
    a = Abs(a) : b = Abs(b) : c = Abs(c) '函數Abs()是取絕對值  
    max = IIf(a > b, a, b)  
    Return IIf(c > max, c, max)  
End Function
```

```
Sub Main()  
    Dim A, B, C As Integer  
    Console.Write("Please input number A:")  
    A = Console.ReadLine()  
    Console.Write("Please input number B:")  
    B = Console.ReadLine()  
    Console.Write("Please input number C:")  
    C = Console.ReadLine()  
    Console.WriteLine("Max of absolute value is: " & maxABS(A, B, C))  
    Console.Read()  
End Sub
```



## Ch6. 習題

---

### ▶ 6-3說明：

- ▶ 1. 使用數學函數需先引入System.Math方法。
- ▶ 要寫在程式模組最前面。

```
Imports System.Math
```

```
Module Module1
```

```
    Function maxABS(a, b, c)
```

```
        :
```

```
    End Function
```

```
    Sub Main()
```

```
        :
```

```
    End Sub
```

```
End Module
```

## Ch6. 習題

---

- ▶ 6-4. 在主程式中接受使用者輸入首項 $a_1$ ，公比 $r$ 與項數 $n$ ，呼叫一副程式並將 $a_1$ 、 $r$ 、 $n$ 的值傳入，在副程式中計算等比級數第 $n$ 項的值並回傳，在主程式中印出該值。
- ▶ 6-5. 在主程式中接受使用者輸入一個陣列的值，將陣列的值與陣列的大小傳入一副程式中，此副程式將會計算該陣列之中位數並回傳，主程式在收到此副程式的回傳值之後印出。



## Ch6. 習題

---

### ► 6-4. 參考程式：

```
Function progression(a1, r, n)
    Return a1 * (r ^ (n - 1))
End Function

Sub Main()
    Dim A1, R, N As Integer
    Console.Write("請輸入首項:") : A1 = Console.ReadLine()
    Console.Write("請輸入公比:") : R = Console.ReadLine()
    Console.Write("請輸入項數:") : N = Console.ReadLine()
    Console.WriteLine(progression(A1, R, N))
    Console.Read()
End Sub
```

### ► 等比級數第n項的值為：

$$a_n = a_1 r^{n-1}$$



## Ch6. 習題

---

### ► 6-5. 參考程式：

```
Function median(ByRef x() As Integer, ByVal L As Integer)
    Array.Sort(x)      '先將陣列排序
    If L Mod 2 <> 0 Then '使用Math.方法的方式就不需要引入了System.Math
        Return x(Math.Ceiling(L / 2))
    Else
        Return (x(Math.Ceiling(L / 2) + 1) + x(Math.Ceiling(L / 2))) / 2
    End If
End Function          'Ceiling()函數是無條件進位

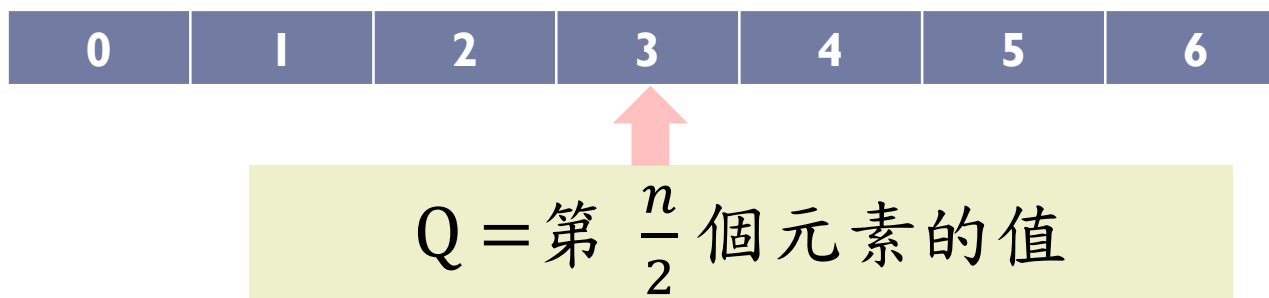
Sub Main()
    Dim L, X() As Integer
    Console.Write("請輸入陣列長度:") : L = Console.ReadLine()
    ReDim X(L)
    For i = 0 To L - 1
        Console.Write("請輸入數值" & i & ": ")
        X(i) = Console.ReadLine()
    Next
    Console.WriteLine("中數:" & median(X, L))
    Console.Read()
End Sub
```



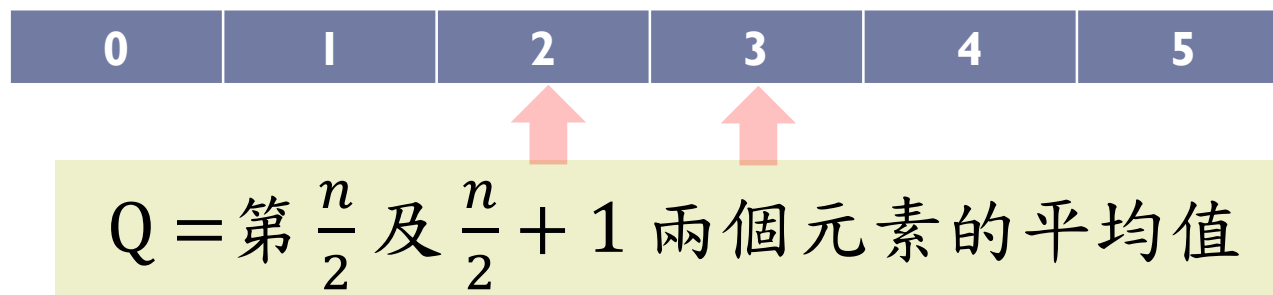
## Ch6. 習題

---

- ▶ 6-5. 說明：
- ▶ 若陣列元素個數為**奇數**，則中位數為正中間元素的值。



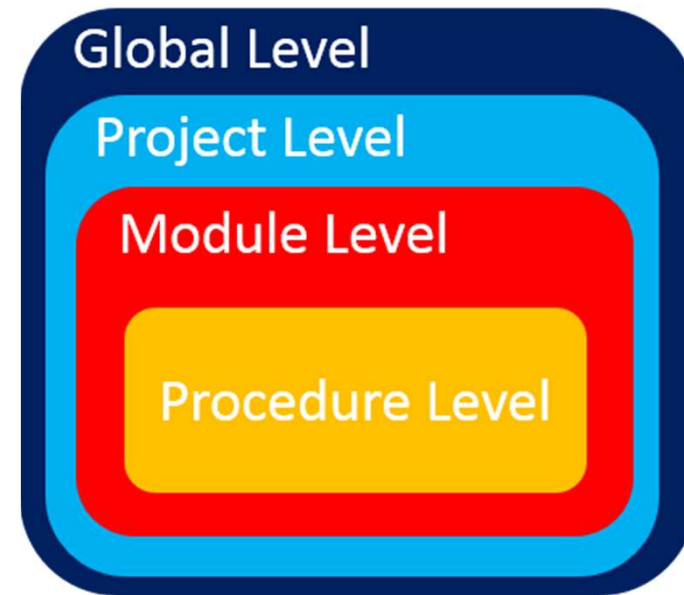
- ▶ 若陣列元素個數為**偶數**，則中位數為中間兩個元素的平均值。



# 變數的可見範圍及生命週期

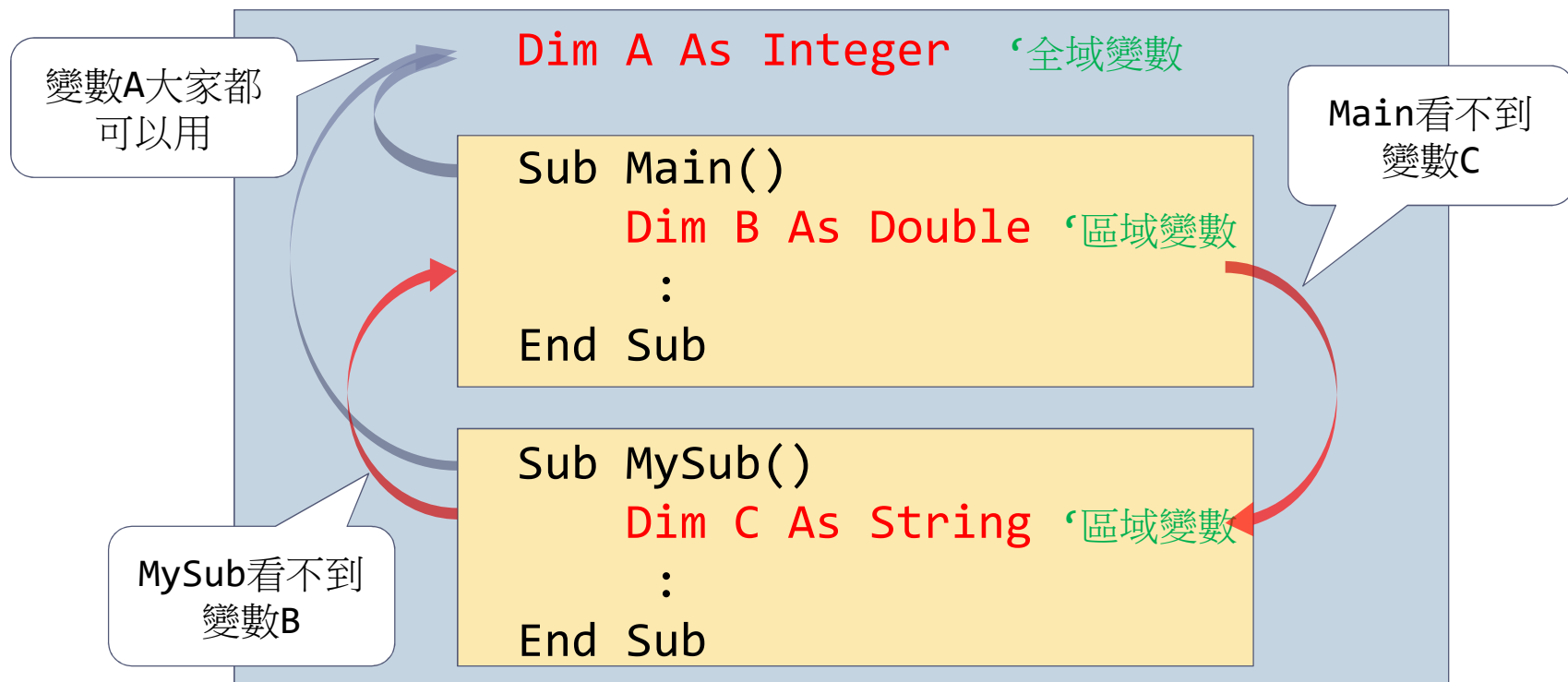
---

- ▶ 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

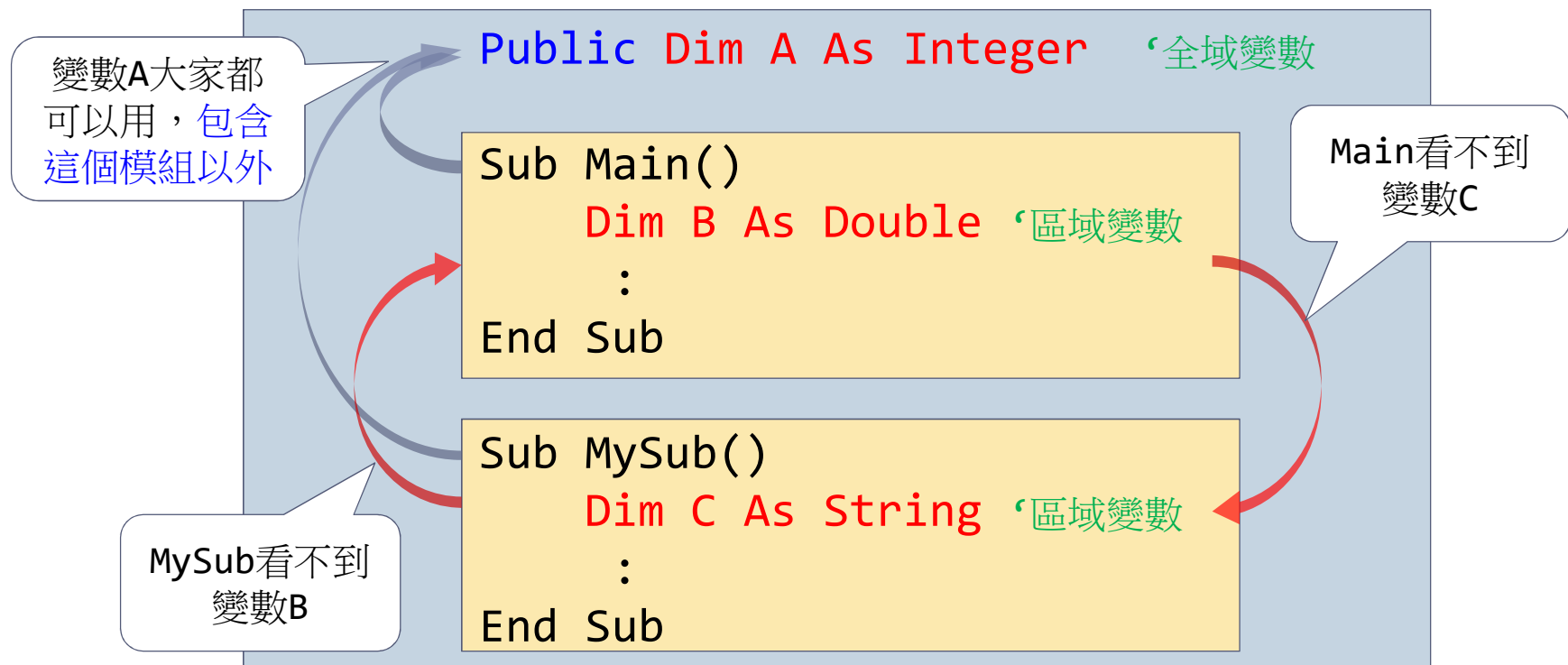
- ▶ 全域變數：
  - ▶ 1.宣告在所有段落之外的變數
  - ▶ 2.Public修飾字可讓變數跨模組





# 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

ConsoleApplication1 - Microsoft Visual Basic 2010 Express (系統管理員)

檔案(F) 編輯(E) 檢視(V) 專案(P) 偵錯(D) 資料(A) 工具(T) 視窗(W) 說明(H)

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

休息一下~

---

