

Ch05流程控制-重複結構

程式設計之基本組成

- 程式是由**循序**、**選擇**和**重複**結構三種基本邏輯架構組成。
- 「**循序結構**」是程式的最基本結構，是**由上到下依序**逐行執行敘述構成。
- 「**選擇結構**」會依照條件，分別執行不同流程的敘述區段。
- 設計程式時，常碰到需將某個程式區段**重複執行**多次，此時就需要「**重複結構**」來完成。
- 「**重複結構**」俗稱為「**迴圈**」（Loop），常用的重複結構有**For...Next**和**Do...Loop**兩種迴圈。

Ch05 內容

- 5.2 For ... Next 迴圈
- 5.3 巢狀 For ... Next 迴圈
- 5.4 Do ... Loop 迴圈
- 5.5 儲存格寫入公式和函數

5.2 For...Next迴圈

- 程式中若有某個「敘述區段」需要反覆執行指定的次數時，就可以使用For...Next迴圈來完成。

語法：

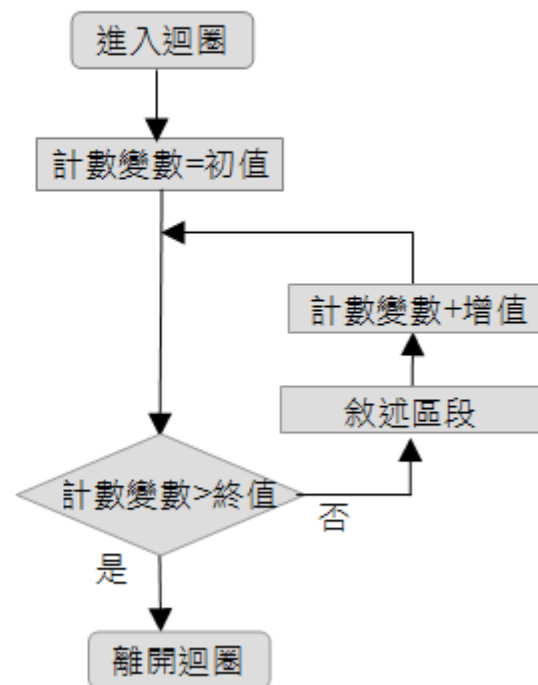
```
For 計數變數 = 初值 To 終值 [Step 增值]
```

```
    ⋮  
    敘述區段
```

```
    [Exit For]
```

```
Next
```

流程圖：



For 計數變數 = 初值 To 終值 [Step 增值]

⋮
敘述區段

[Exit For]

Next

- For...Next迴圈中的**計數變數**必須是**數值**資料型別的變數，而**初值**、**終值**和**增值**則可以為**數值變數**、**數值常值**或**數值運算式**。
- **增值**不能為**零**，若省略不寫則**預設增值為1**。
- 若**初值小於終值**，增值必為正值，計數變數大於終值時離開迴圈；
- 若**初值大於終值**，增值必為負值，計數變數小於終值時離開迴圈。
- 如要提早離開For...Next迴圈，可用**Exit For**敘述。
- For ...Next迴圈內的敘述，應按<Tab>鍵來讓敘述區段**內縮**以方便閱讀。

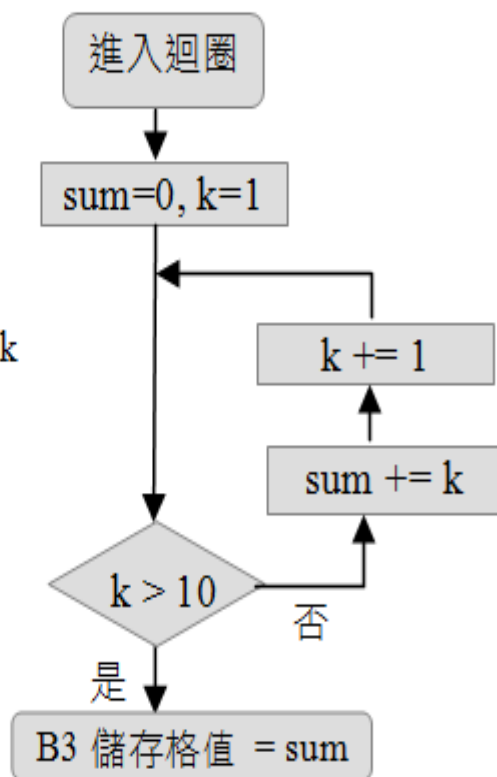
● 例

將 $1 + 2 + 3 + \dots + 10$ 的總和存入 B3 儲存格。

① 流程圖

假設：

1. 計數變數 k
2. 初值 = 1
3. 終值 = 10



② 程式碼

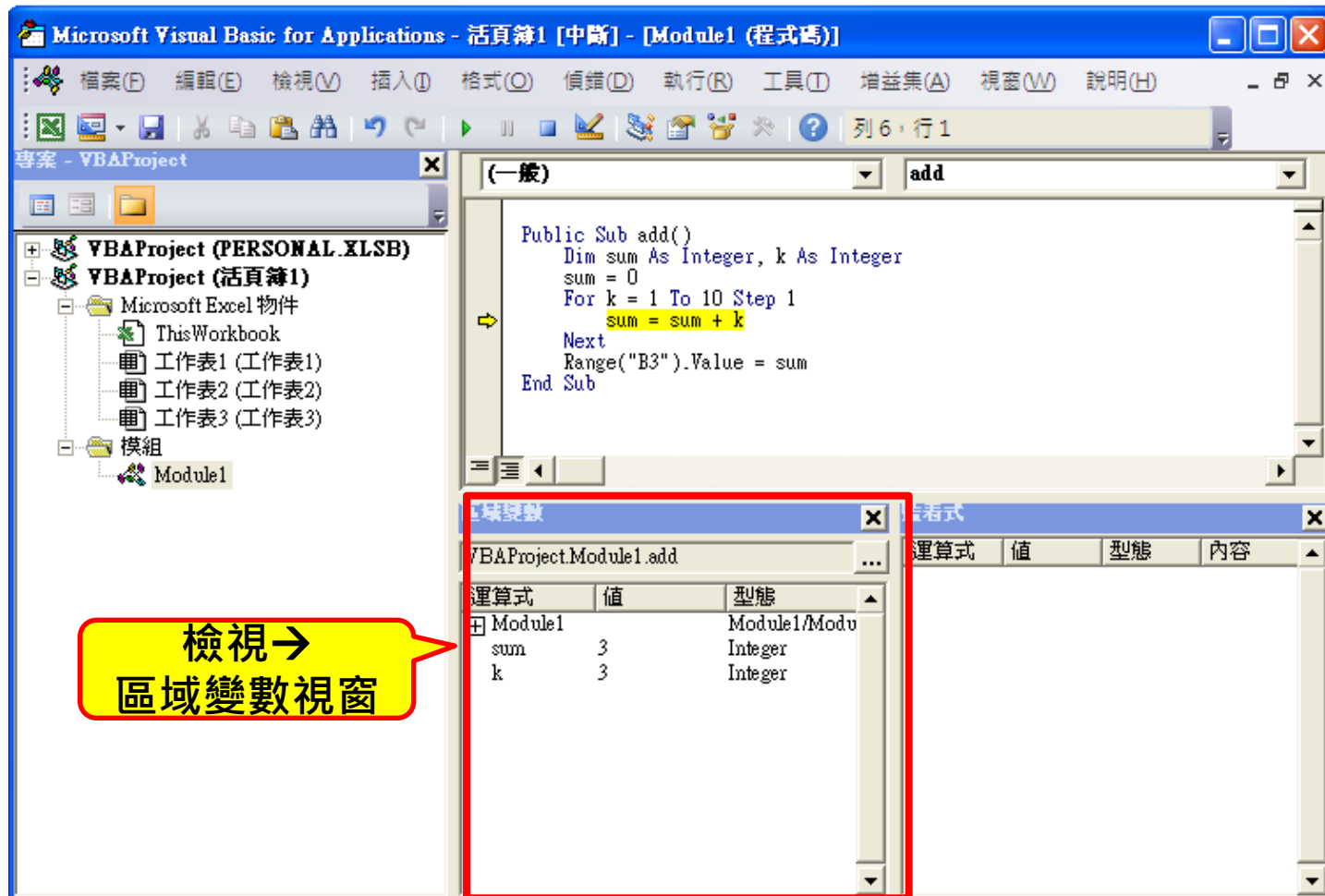
```
Dim k As Integer, sum As Integer  
sum = 0  
For k = 1 To 10 Step 1  
    sum = sum + k  
Next  
Range("B3").Value = sum
```

執行步驟：

1. 插入→模組
2. 插入→程序add
3. 輸入程式碼
4. 游標放置在Sub add()中
5. [F8]: 偵錯→逐行

程式碼：

```
Dim sum As Integer, k As Integer
sum = 0
For k = 1 To 10 Step 1
    sum = sum + k
Next
Range("B3").Value = sum
```



► 隨堂練習

1. 試將 $11 + 9 + \dots + 3 + 1 = ?$ 的運算式，使用 For...Next 迴圈敘述來撰寫程式？
 2. 試將 $0.5 + 1.0 + \dots + 4.5 + 5 = ?$ 的運算式，使用 For...Next 迴圈敘述來撰寫程式？
- 可以透過區域變數視窗來觀察 For...Next 迴圈的運作流程，以及相關變數的變化情形。

```
Public Sub add1()  
    Dim sum As Integer  
    Dim k As Integer  
    sum = 0  
    For k = 11 To 1 Step -1  
        sum = sum + k  
    Next  
    Range("B3").Value = sum  
End Sub
```


```
Public Sub add2()  
    Dim sum As Double  
    Dim k As Double  
    sum = 0  
    For k = 0.5 To 5 Step 0.5  
        sum = sum + k  
    Next  
    Range("B3").Value = sum  
End Sub
```





實作 FileName : For.xlsm

設計一個迴圈測試程式，可輸入 For...Next 迴圈的初值、終值和增值，執行後會計算出總和並顯示。

► 輸出要求

	A	B	C
1	初值(整數)：	1	
2	終值(整數)：	100	
3	增值(整數)：	1	
4	總和：	5050	

	A	B	C
1	初值(整數)：	100	
2	終值(整數)：	1	
3	增值(整數)：	-5	
4	總和：	1050	

► 解題技巧

- 從儲存格讀取使用者輸入的初值、終值和增值，並分別存放在start_num(初值)、end_num(終值)和step_num (增值)變數中。
- 將start_num(初值)、end_num(終值)和step_num (增值)變數，代入For...Next迴圈中計算出總和。

```
For i = start_num To end_num Step step_num  
    sum += i  
Next
```

Step ③ 編寫程式碼

FileName: For.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()  
02     Dim start_num, end_num, step_num, sum As Integer  
03     start_num = Range("B1").Value  
04     end_num = Range("B2").Value  
05     step_num = Range("B3").Value  
06     sum = 0                '預設總和為 0  
07     For i = start_num To end_num Step step_num  
08         sum = sum + i  
09     Next  
10     Range("B4").Value = sum  
11 End Sub
```

► 隨堂測驗

將上面實作增加增值的檢查，以避免程式執行時的錯誤。第一增值不能為零、第二當終值大於初值時，增值必須大於零、第三終值小於初值時，增值必須小於零。

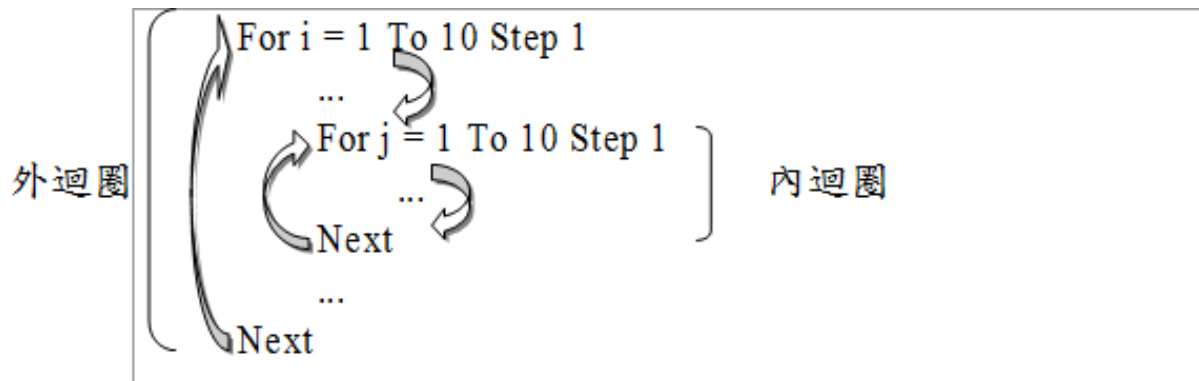
	A	B	C
1	初值(整數):	1	
2	終值(整數):	10	執行
3	增值(整數):	0	
4	總和:	增值不能為零	

	A	B	C	D	E
1	初值(整數):	1			
2	終值(整數):	10	執行		
3	增值(整數):	-1			
4	總和:	增值小於零時，終值必須小於初值			

	A	B	C	D	E
1	初值(整數):	10			
2	終值(整數):	1	執行		
3	增值(整數):	1			
4	總和:	增值大於零時，終值必須大於初值			

5.3 巢狀For...Next迴圈

- 巢狀For...Next迴圈就是指一個For...Next迴圈內的敘述區段，還有For...Next敘述區段。
- 先進外迴圈將內迴圈的敘述區段執行一次，再回到外迴圈比較計數變數和終值是否超出範圍？
- 要特別注意每個For都必須有對應的Next，迴圈彼此之間是不允許相互交錯。



● 例

每列都顯示五個 “*” 星號，共三個水平列。可以將r設為外部迴圈的計數變數，c為內部迴圈的計數變數。

- Step ① 設定 r 變數由 1~3 表水平列數，c 變數由 1 ~ 5 表每列印出的個數。
- Step ② r = 1 即第 1 列時，c = 1 ~ 5 執行 5 次，每次設儲存格 Cells(1, c)的值為“*”。
- Step ③ r = 2 即第 2 列時，c = 1 ~ 5 執行 5 次，每次設儲存格 Cells(2, c)的值為“*”。
- Step ④ r = 3 即第 3 列時，c = 1 ~ 5 執行 5 次，每次設儲存格 Cells(3, c)的值為“*”。

```
For r = 1 To 3
  For c = 1 To 5
    Cells(r, c).Value = "*"
  Next
Next
```

【輸出結果】

	A	B	C	D	E
1	*	*	*	*	*
2	*	*	*	*	*
3	*	*	*	*	*



實作 FileName : One2Six.xlsm

設計一個可以顯示如下圖的 1 至 6 漸增的程式。


► 輸出要求

	A	B	C	D	E	F
1	1					
2	1	2				
3	1	2	3			
4	1	2	3	4		
5	1	2	3	4	5	
6	1	2	3	4	5	6

► 解題技巧

- 列(橫向)和欄(縱向)都是由1到6，因此可以使用巢狀迴圈。外圈 r 為水平列(橫向)執行6次($r=1\sim 6$)，內圈 c 為垂直欄(縱向)由1到6漸增，所以要將內圈的終值設為 r 。

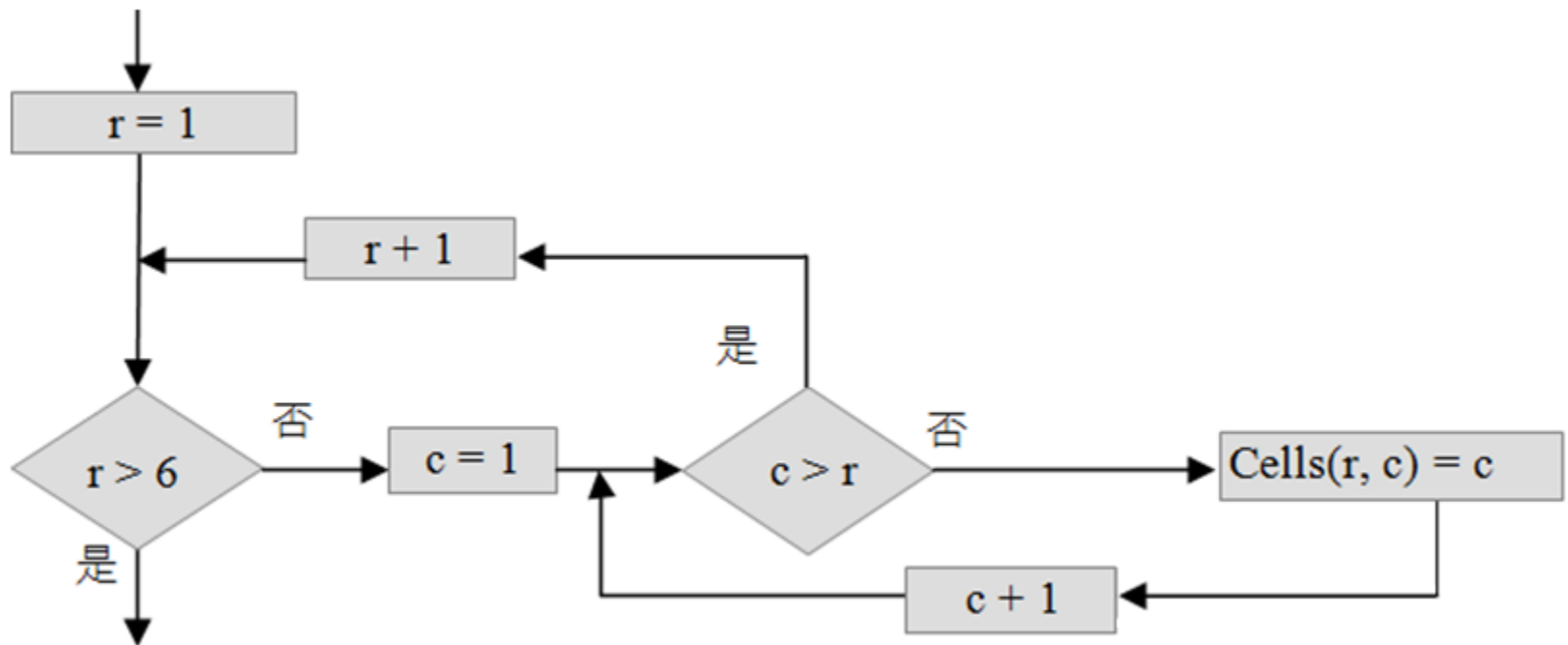
$r=1$:	$c=1$	To	1	\Rightarrow	顯示	c 值 1 次
$r=2$:	$c=1$	To	2	\Rightarrow	顯示	c 值 2 次
$r=3$:	$c=1$	To	3	\Rightarrow	顯示	c 值 3 次
$r=4$:	$c=1$	To	4	\Rightarrow	顯示	c 值 4 次
$r=5$:	$c=1$	To	5	\Rightarrow	顯示	c 值 5 次
$r=6$:	$c=1$	To	6	\Rightarrow	顯示	c 值 6 次



► 解題技巧

- 儲存格以Cells方式來指定會比較簡便。
- Cells()內第一個參數是指定列數，第二個參數是指定欄數。

例如：Cells(3, 2)就等於B3儲存格。



Step 3 編寫程式碼

FileName: One2Six.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()
```

```
02     For r = 1 To 6
```

```
03         For c = 1 To r
```

```
04             Cells(r, c).Value = c
```

```
05         Next
```

```
06     Next
```

```
07 End Sub
```

► 隨堂測驗

設計一個可以顯示如右圖的 1 至 6 漸減的程式。

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	1	2	3	4	5	
3	1	2	3	4		
4	1	2	3			
5	1	2				
6	1					

執行


5.3.2 取得工作表資料的最下列和最右欄

- 要處理Excel工作表中的資料時，常常需要知道最下一列資料的列號，以及最右一欄資料是第幾欄，以便設定For ... Next迴圈的終值。
- 使用儲存格的**End屬性**，由A1儲存格開始沿**xlDown**方向找到最下一列，再配合**Row屬性**取得列號。
- 沿**xlToRight**方向找到最右一欄，再配合**Column屬性**取得是第幾欄。
- 例如：要取得工作表最下一列的列號，和最右一欄是第幾欄，程式寫法為：

```
rNum = Range("A1").End(xlDown).Row  
cNum = Range("A1").End(xlToRight).Column
```



實作 FileName : BackColor.xlsm

在公司存貨月報表上按  鈕，會自動檢查表格的範圍，然後每隔一行填入紫色作為底色，以方便閱讀表格。

► 輸出要求

	A	B	C	D	E	F	G
1	存貨名稱	月初數量	本月進貨數量	本月出貨數量	本月結存數量		
2	巧克力棒隨手杯	1000	500	856	644	填底色	
3	草莓夾心餅乾	2500	1200	2138	1562		
4	巧克力餅乾	1200	400	941	659		
5	純麥蘇打餅乾	800	300	514	586		
6	健康蔬果脆片	600	200	437	363		
7	蔓越莓乾顆粒	1500	400	1345	555		
8	瑞士牛乳餅	700	100	549	251		
9	鮮蝦風味脆餅	400	200	462	138		
10	巧克力酥片	900	300	1098	102		
11							

► 解題技巧

- 使用**End屬性**取得表格的最下列列號，以及最右欄是第幾欄。
- 因為是隔列填底色，所以For迴圈的**增值**要設為**2**。
- 儲存格範圍可以使用**Range(Cells(列1,欄1), Cells(列2,欄2))**來表示，其中Cells(列1,欄1)為左上角儲存格、Cells(列2,欄2) 為右下角儲存格。
- 使用儲存格的**Interior.Color屬性**可以設定底色。

Step 3 編寫程式碼

FileName: BackColor.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()  
02     Dim rNum As Integer  
03     Dim cNum As Integer  
04     rNum = Range("A1").End(xlDown).Row '取得最下列是第幾列  
05     cNum = Range("A1").End(xlToRight).Column '取得最右欄是第幾欄  
06     For r = 2 To rNum Step 2          '從第二列開始隔列填色  
07         Range(Cells(r, 1), Cells(r, cNum)).Interior.Color = RGB(255, 0, 255)  
08     Next  
09 End Sub
```

► 隨堂測驗




將上面實作改成隔欄填入底色。

	A	B	C	D	E	F	G
1	存貨名稱	月初數量	本月進貨數量	本月出貨數量	本月結存數量	填底色	
2	巧克力棒隨手杯	1000	500	856	644		
3	草莓夾心餅乾	2500	1200	2138	1562		
4	巧克力餅乾	1200	400	941	659		
5	純麥蘇打餅乾	800	300	514	586		
6	健康蔬果脆片	600	200	437	363		
7	蔓越莓乾顆粒	1500	400	1345	555		
8	瑞士牛乳餅	700	100	549	251		
9	鮮蝦風味脆餅	400	200	462	138		
10	巧克力酥片	900	300	1098	102		
11							




實作

FileName : Print.xlsm

本實作學習如何修改巨集產生的程式碼變成符合本實作的程式碼。本實作要求將工作表中 1~10 個班級所有同學的六科成績，按  鈕就會將 1~10 個班級，逐班顯示所有同學的六科成績。建議先透過功能區「資料」標籤頁的「自動篩選」鈕，點按班級的  下拉鈕，如下圖選擇第一班，便會由成績工作表中篩選出並顯示第一班所有同學的六科成績，將此過程錄製成巨集，再將巨集自動產生的程式碼複製給  按鈕 Command Button1 的 Click 事件程序中，然後在 Click 事件程序中修改相關程式碼，變成可列印 1-10 個班級所有同學六科成績。

► 輸出要求

	A	B	C	D	E	F	G	H	I	J	K
1	班級	座號	姓名	程式設計	計算機概論	會計學	經濟學	記帳實務	專題製作		
2	1	1	楊紋	64	73	89	78	44	80		
3	1	2	怡臻	84	85	80	88	86	98		
4	1	3	蔣庭	84	90	60	92	72	95		
5	1	4	梁嬌	78	74	49	82	58	93		

► 解題技巧

- 自動篩選和列印的程式碼，我們可以使用錄製巨集的方式將程式碼錄下。操作步驟如下：

- ① 錄製巨集。
- ② 執行自動篩選功能。
- ③ 只顯示1班同學的成績資料。
- ④ 執行【檔案/列印】指令。
- ⑤ 關閉自動篩選功能。
- ⑥ 停止錄製巨集。



► 解題技巧

- 將錄製的程式碼複製到 CommandButton1_Click 事件程序中，下面就是所錄製的巨集程式。

```
01 Range("A1").Select      '選取 A1 儲存格
02 Selection.AutoFilter    '開啟自動篩選
03 ActiveSheet.Range("$A$1:$I$102").AutoFilter Field:=1, Criteria1:="1"  '篩選條件為 1 班
04 ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _
    IgnorePrintAreas:=False '列印 1 份
05 Selection.AutoFilter    '關閉自動篩選
```

► 解題技巧

- 錄製完成的巨集程式，可以再**加上VBA**的語法來增加其功能。錄製巨集是只列印一個班的程式，可以加上For ... Next迴圈來列印出十個班的成績表。
- 第三行程式**Criteria1:="1"**，是設定篩選條件為"1"，所以必須用**Str函數**將**計數變數**轉為**字串型態**。
- Range("\$A\$1:\$I\$102")指定篩選的儲存格範圍，**修改為Range("A:I")**。
- 執行【移除Module1】指令刪除錄製的巨集程式。
- 命令按鈕的**PrintObject**屬性值設為**False**，**是指定**列印時不印出控制項。

Step 3 編寫程式碼

FileName: Print.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()  
02     Range("A1").Select      '選取 A1 儲存格  
03     Selection.AutoFilter    '開啟自動篩選  
04     For i = 1 To 10  
05         ActiveSheet.Range("A:I").AutoFilter Field:=1, Criteria1:=Str(i) '篩選條件為 i 字串  
06         ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True, _  
07             IgnorePrintAreas:=False      '列印 1 份  
08     Next  
09     Selection.AutoFilter    '關閉自動篩選  
10 End Sub
```

5.4 Do ... Loop迴圈

- Do ... Loop迴圈沒有計數變數，靠條件式來決定是否離開迴圈，所以迴圈執行次數無法預知。
- 迴圈的條件式置於第一行就稱為「前測式迴圈」；放在最後一行就稱為「後測式迴圈」。
- 「前測式迴圈」要先判斷條件式是否成立？如果成立才執行迴圈內的敘述。如果條件式一開始便不成立，則迴圈內的敘述不會被執行。
- 「後測式迴圈」是先執行迴圈內的敘述後才判斷條件式，所以迴圈內的敘述至少被執行一次。
- 迴圈內必須有使條件式不滿足的敘述，如此才能離開迴圈，否則會反覆執行形成無窮迴圈。

Do While迴圈

前測式迴圈：

- 當While的 **<條件式>** 為True時，執行迴圈內的敘述區段一次，然後再回到迴圈的起點，再重新判斷 **<條件式>**，一直到 **<條件式>** 為False時，才結束。
- 若要**中途離開Do迴圈**，可以使用**Exit Do**敘述。

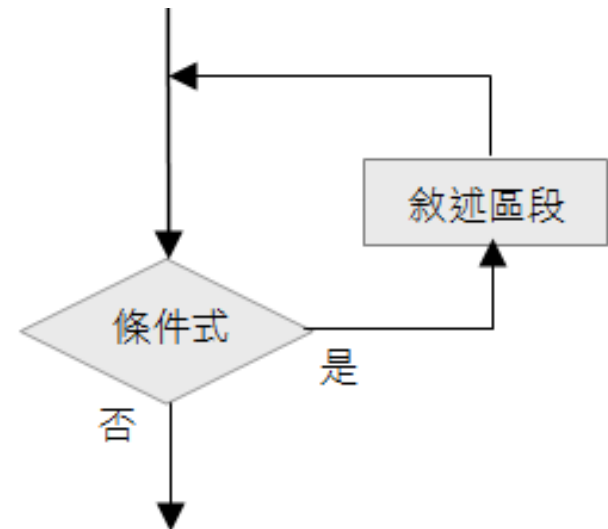
語法：

Do While <條件式>

敘述區段

[Exit Do]

Loop



Do While迴圈

後測式迴圈：

- 先執行迴圈內的敘述區段一次，才判斷 <條件式>。
- 若 <條件式> 為True，會再執行迴圈內的敘述區段一次，直到 <條件式> 為False才結束迴圈。

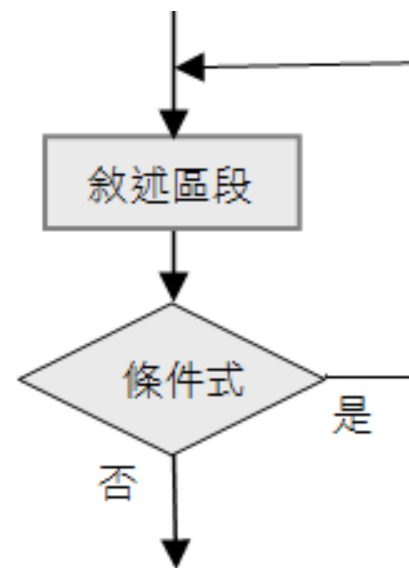
語法：

Do

敘述區段

[Exit Do]

Loop While <條件式>



Do Until 迴圈

Do Until迴圈是 <條件式> 不成立時，才執行迴圈敘述。

前測式迴圈：先判斷後執行

- Do Until...Loop，當<條件式>不成立時，才執行迴圈內的敘述區段；若<條件式>成立，才離開迴圈。

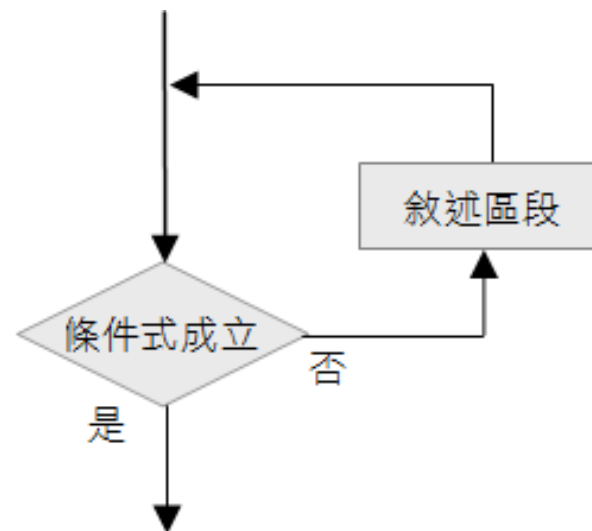
語法：

Do Until <條件式>

敘述區段

[Exit Do]

Loop

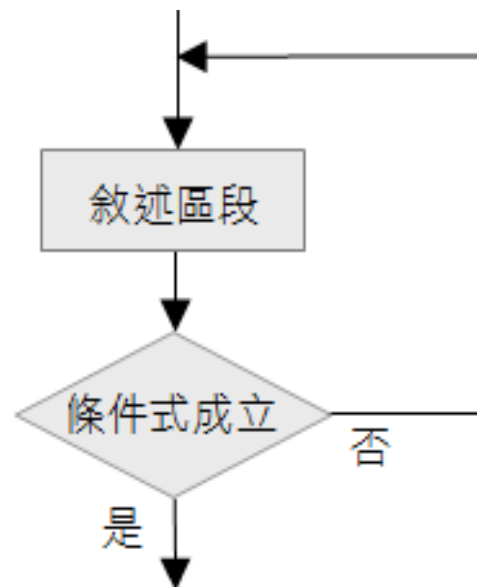


Do Until 迴圈

後測式迴圈：先執行後判斷

- Do...Loop Until，當<條件式>不成立時，才繼續執行迴圈內的敘述區段；若<條件式>成立就離開迴圈。

語法
Do
敘述區段
[Exit Do]
Loop Until <條件式>



- Do...Loop條件式迴圈無論是前測式或後測式，都能達到重複執行效果。下面用Do ... Loop四種不同的迴圈，來撰寫計算 $sum = 1 + 2 + 3 \dots + 10$ 程式：

一、使用Do While迴圈

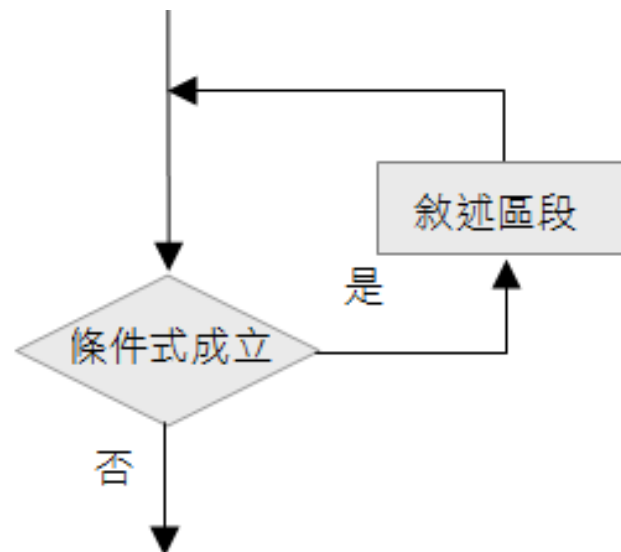
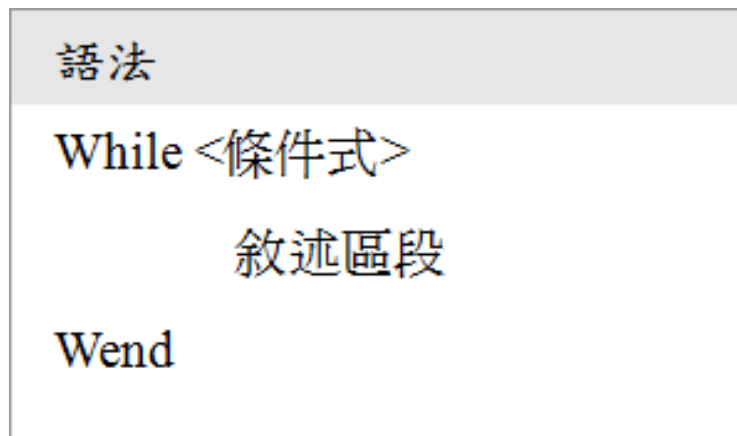
1.使用 Do While Loop	2. 使用 Do Loop While
<pre>Dim i As Integer, sum As Integer i = 0: sum = 0 Do While i < 10 i = i + 1 sum = sum + i Loop</pre>	<pre>Dim i As Integer, sum As Integer i = 0: sum = 0 Do i = i + 1 sum = sum + i Loop While i < 10</pre>

二、使用Do Until迴圈

1. 使用 Do Until Loop	2. 使用 Do Loop Until
<pre>Dim i As Integer, sum As Integer i = 0: sum = 0 Do Until i >= 10 i = i + 1 sum = sum + i Loop</pre>	<pre>Dim i As Integer, sum As Integer i = 0: sum = 0 Do i = i + 1 sum = sum + i Loop Until i >= 10</pre>

While...Wend 迴圈

- While...Wend迴圈和Do While...Loop迴圈類似。
- While...Wend迴圈屬於「前測式迴圈」，當While後面的條件式結果為True時，會將迴圈內的敘述區段執行一次；當條件式結果為False時，才離開迴圈。
- While...Wend迴圈沒有中途離開迴圈的敘述。



● 例

用While...Wend迴圈來計算 $sum = 1 + 2 + 3 \dots + 10$:

```
Dim i As Integer, sum As Integer
```

```
i = 0: sum = 0
```

```
While i < 10
```

```
    i = i + 1
```

```
    sum = sum + i
```

```
Wend
```



實作 FileName : Sum.xlsm

從 A2 儲存格開始逐一向下讀取數值，直到沒有數值為止，計算出的總計在 B2 儲存格中顯示。

► 輸出要求

	A	B	C
1	數值	總計	計算
2	45	81	
3	36		
4			

	A	B	C
1	數值	總計	計算
2	45	258	
3	36		
4	82		
5	71		
6	24		
7			

► 解題技巧

- 因為不知道使用者會輸入多少的數值，所以要使用**前測式Do While... Loop迴圈**，當儲存格內有數值才加入總計中，否則就結束迴圈。
- 使用**IsEmpty()****函數**來檢查儲存是否為空白，如果傳回值為True表為空白；傳回值為False表有資料。
- 因為Do ... Loop迴圈不像For迴圈會自動增值，所以要在迴圈內寫程式將**變數加一**，來讀取下一個儲存格的數值。

Step 3 編寫程式碼

FileName: Sum.xlsm (工作表 1 程式碼)

01 Sub test()

02 Dim i, sum As Integer

03 i = 2: sum = 0

04 Do While IsEmpty(Cells(i, 1)) = False

05 sum = sum + Cells(i, 1).Value

06 i = i + 1

07 Loop

08 Range("B2").Value = sum

09 End Sub

► 隨堂測驗

將上面實作改成向右讀取儲存格數值，並改用 Do Until... Loop 前測式迴圈來編寫程式。

	A	B	C
1	數值	45	36
2	總計	81	計算

	A	B	C	D	E	F	G
1	數值	45	36	82	71	24	
2	總計	258	計算				

無窮迴圈

- 如果Do...Loop迴圈不加入While或Until等任何條件式，就稱為「無窮迴圈」。
- 因為程式會不斷地執行迴圈不停止，所以必須使用Exit Do敘述來中途離開Do迴圈。
- 若因條件式邏輯不對，造成無窮迴圈無法停止程式時，可以按 Ctrl + Break 鍵來中止程式的執行。

語法：

```
Do
  ...
  If <條件式> Then Exit Do
  ...
Loop
```

● 例

用Do...Loop迴圈來計算 $\text{sum} = 1 + 2 + 3 \dots + 10$ 。

```
Dim i As Integer, sum As Integer
i = 0: sum = 0
Do
    i = i + 1
    sum = sum + i
    If i = 10 Then Exit Do
Loop
```



實作 FileName : Check Score.xlsm

檢查 B2 到 F4 儲存格範圍中是否有輸入資料，如果檢查到某儲存格沒有輸入資料，就用 MsgBox 函數顯示哪個儲存格沒有成績。如果所有的儲存格都有資料，就用 MsgBox 函數顯示成績輸入完畢！

► 輸出要求

	A	B	C	D	E	F	G
1	座號	國文	英語	數學	物理	歷史	
2	1	99		77	66	55	檢查
3	2		70	80	90	100	
4	3	97	96	95		93	
5							
6							

Microsoft Excel

\$C\$2儲存格沒有成績

確定

	A	B	C	D	E	F	G
1	座號	國文	英語	數學	物理	歷史	
2	1	99	88	77	66	55	檢查
3	2	60	70	80	90	100	
4	3	97	96	95	94	93	
5							
6							

Microsoft Excel

成績檢查完畢！

確定

► 解題技巧

- 因為是矩形的儲存格範圍，所以要使用**巢狀Do ... Loop迴圈**來逐格檢查。
- 因為Do ... Loop為無窮迴圈，所以一定要有**Exit Do敘述**，才能跳離迴圈。
- 使用**IsEmpty()****函數**檢查是否有輸入資料，當傳回值為True表沒有資料；False則表有資料。
- 使用Cells()的**Address屬性**，可以取得該儲存格的位址，例如B2儲存格位址為「\$B\$2」。
- 檢查到有儲存格沒有輸入資料，用**MsgBox**顯示訊息後，就用**End敘述**來停止程式的執行。

Step 3 編寫程式碼

FileName: CheckScore.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()  
02     Dim r As Integer    '列  
03     Dim c As Integer    '欄  
04     r = 1    '設 r=1  
05     Do  
06         r = r + 1    '列加 1  
07         If r = 5 Then Exit Do    '到第 5 列就離開迴圈  
08         c = 1    '設 c=1  
09         Do  
10             c = c + 1    '欄加 1  
11             If c = 7 Then Exit Do    '到第 7 欄就離開迴圈  
12             If IsEmpty(Cells(r, c)) Then    '如果 IsEmpty()傳回值為 True 表沒有資料  
13                 '用 MsgBox 顯示沒有成績的儲存格位址  
14                 MsgBox (Cells(r, c).Address & "儲存格沒有成績")  
15                 End    '結束程式  
16             End If  
17         Loop  
18     Loop  
19     MsgBox ("成績輸入完畢！")
```

► 隨堂測驗

將上面實作改成用 For...Next 迴圈檢查，並增加檢查輸入的數值是否介於 0~100，以及是否為數值。

	A	B	C	D	E	F	G
1	座號	國文	英語	數學	物理	歷史	
2	1	99	88	177	66	55	檢查
3	2	60	70	80	90	100	
4	3	97	96	95	94	93	
5							
6							

Microsoft Excel

SDS2儲存格成績超出範圍

確定

	A	B	C	D	E	F	G
1	座號	國文	英語	數學	物理	歷史	
2	1	99	88	77	66	55	檢查
3	2	60	七十	80	90	100	
4	3	97	96	95	94	93	
5							
6							

Microsoft Excel

SCS3儲存格不是數值

確定

5.5 儲存格寫入公式和函數

- 在VBA中可以將公式和函數寫入儲存格中，而且可以透過變數提高程式的彈性。
- 在VBA環境中要將公式和函數寫入儲存格時，可以使用**Formula**或**Value**屬性，屬性值要將公式和函數改為**字串資料型態**。
- 例如在C1儲存格寫入A1儲存格值乘於B1儲存格值公式，寫法為：

```
Range("C1").Formula = "=A1*B1"    '或 Range("C1").Value = "=A1*B1"
```

- 例如：在E1儲存格寫入A1到D1儲存格值的總和函數，程式寫法為：

```
Range("E1").Formula = "=SUM(A1:D1)"    '或 Range("E1").Value = "=SUM(A1:D1)"
```

- 配合迴圈時列號可以使用變數和&運算子，就可以大批寫入公式。上面兩個簡例可以改寫為：

```
Dim r As Integer
```

```
Range("C" & r).Formula = "=A" & r & "*B" & r    ' 若 r=1 時 C1 ⇐ =A1*B1
```

```
Range("E" & r).Formula = "=SUM(A" & r & ":D" & r & ")" ' 若 r=1 時 E1 ⇐ =SUM(A1:D1)
```

- 寫入公式時如果希望欄名也可以變動，此時儲存格可以改用[R1C1]欄名列號表示法比較方便。
- 可以使用儲存格的FormulaR1C1或Value屬性，來指定公式或函數。
- [R1C1]欄名列號表示法是代表儲存格的位置，R後面是儲存格所在的水平列號，C後面是指在第幾垂直欄(欄號)，
- 例如：B3儲存格為R3C2(第三列第二欄)。
- [R1C1]欄名列號表示法分成絕對參照、混和參照和相對參照三種方式。

5.5.1 絕對參照

- [R1C1]欄名列號表示法如果**明確指定列號和欄號**時，就稱為**絕對參照**。
- 例如**D5**儲存格為**R5C4**，在Excel中會顯示成\$D\$5。
- 例如：在C1儲存格寫入A1儲存格值乘於B1儲存格值公式，程式寫法為：

```
Range("C1").FormulaR1C1 = "=R1C1*R1C2"
```

'使用變數時

```
Dim r As Integer, c As Integer
```

```
r = 1 : c = 3
```

```
Cells(r, c).FormulaR1C1 = "=R" & r & "C" & c-2 & "*R" & r & "C" & c-1
```

5.5.2 混和參照

- [R1C1]欄名列號表示法如果只指定列號或欄號時，就稱為混和參照。
- 如果要指定同一列或同一欄的儲存格，就可以使用混和參照比較簡潔。
- 例如：在E1儲存格寫入A1到D1儲存格值的總和函數，程式寫法為：

```
Range("E1").FormulaR1C1 = "=SUM(RC1:RC4)"
```


5.5.3 相對參照

- [R1C1]欄名列號表示法可以用**位移**來指定儲存格，此時就稱為**相對參照**。
- 位移值用**中括號[]**框住，**R**的位移值為**正表向下**、為**負表向上**；**C**的位移值為**正表向右**、為**負表向左**。
- 例如：在**B2儲存格**使用「**=R[1]C**」，表指定**B3儲存格(下移一行)**；**B2儲存格**使用「**=R[-1]C[2]**」，表指定**D1儲存格(上移一行、右移兩欄)**。
- 例如在**A4儲存格**寫入**A1到A3儲存格**值的總和函數，程式寫法為：


```
Range("A4").FormulaR1C1 = "=SUM(R[-3]C:R[-1]C)"
```




實作 FileName : Total.xlsm

按  鈕後，會從第二列起檢查是否有資料，若有就在 D 欄填入公式，例如 D2 儲存格公式為「=B2*C2」。檢查到沒有資料時，就在該列的 C 欄填入「總計：」、D 欄填入公式「=SUM(D2:Dr)」，公式中的 r 為空白列的列號。

► 輸出要求

	A	B	C	D	E	F
1	品名	數量	單價	小計		
2	深層卸粧精華露	2	265			
3	保濕洗面乳	5	199			
4	玫瑰香氛沐浴乳	3	99			
5	保濕洗髮精	1	365			
6						

	A	B	C	D	E	F
1	品名	數量	單價	小計		
2	深層卸粧精華露	2	265	530		
3	保濕洗面乳	5	199	995		
4	玫瑰香氛沐浴乳	3	99	297		
5	保濕洗髮精	1	365	365		
6		總計：		2187		

► 解題技巧

- 在VBA中可以為儲存格寫入Excel的公式和函數，列號可以變數並用&運算子連接。
- 因為不知道會有多少筆資料，所以使用Do While... Loop迴圈。
- 當A欄儲存格的值不等於空字串時，就在D欄填入公式。
- 當A欄儲存格的值等於空字串時，就會離開Do ... Loop迴圈，此時在C和D欄填入資料。

Step 3 編寫程式碼

FileName: Total.xlsm (工作表 1 程式碼)

```
01 Private Sub CommandButton1_Click()  
02     Dim r As Integer  
03     r = 2    '從第 2 列開始  
04     Do While Cells(r, 1) <> ""  
05         Cells(r, 4).Formula = "=B" & r & "*C" & r '設值為公式=Br*Cr  
06         r = r + 1    '列數加 1  
07     Loop  
08     Cells(r, 3).Value = "總計："   
09     Cells(r, 4).Formula = "=SUM(D2:D" & r - 1 & ")" '設值為公式=SUM(D2:Dr)  
10 End Sub
```

► 隨堂測驗

將上面實作填入的公式，儲存格改用[R1C1]
欄名列號表示法。