

# Visual Basic 程式設計

- ◆ 字串函數
- ◆ 鍵盤事件
- ◆ ActiveX控制項 – MaskedTextBox元件

# 字串函數

◆ 字串變數: `str = "Abc123"`

■ 轉為小寫字串: `LCase(字串)`

◆ `LCase(str) → "abc123"`

■ 轉為大寫字串: `UCase(字串)`

◆ `UCase(str) → "ABC123"`

■ 計算字串長度: `Len(型別)`

◆ `Len(str) = 6`

```
Dim A As Integer
```

```
A=123
```

```
Print Len(A)
```

2

# 字串函數 (cont.)

## ◆ 子字串搜尋：

- InStr([搜尋的位置],字串1,字串2[,比較法])
- 傳回字串2在字串1出現的位置
  - ◆ 找不到則傳回0
- InStr("abcdecdecd","cd") → 3
- InStr(4,"abcdecdecd","cd") → 6
- 比較法：
  - ◆ vbBinaryCompare → 分大小寫
  - ◆ vbTextCompare → 不分大小寫

# 字串函數 (cont.)

- 使用比較法時，起始位置要輸入
- InStr(1, "abcd", "Cd", vbBinaryCompare) → 0
- InStr(1, "abcd", "Cd", vbTextCompare) → 3

◆ 擷取子字串: Mid(字串, 起始位置[, 長度])

- Mid("test test", 6) → "test"
- Mid("test test", 6, 3) → "tes"

# 字串函數 (cont.)

◆ 從左邊取子字串:Left(字串,長度)

■ Left("test1 test2",5)→"test1"

◆ 從右邊取子字串:Right(字串,長度)

■ Right("test1 test2",5)→"test2"

◆ 去除左右空白字元:Trim(字串)

■ Trim(" test ")→"test"

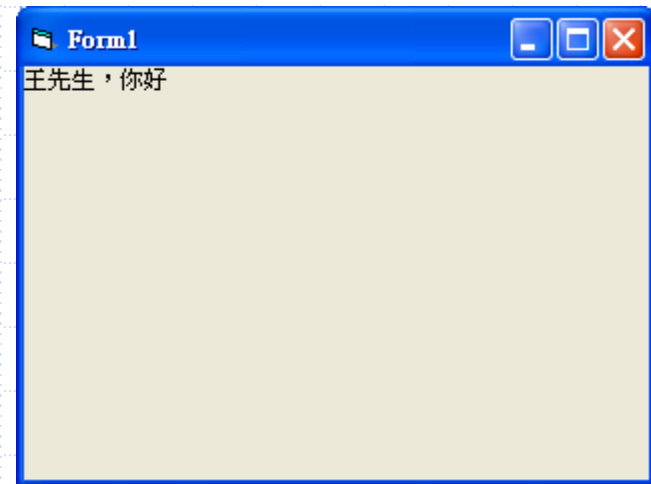
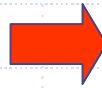
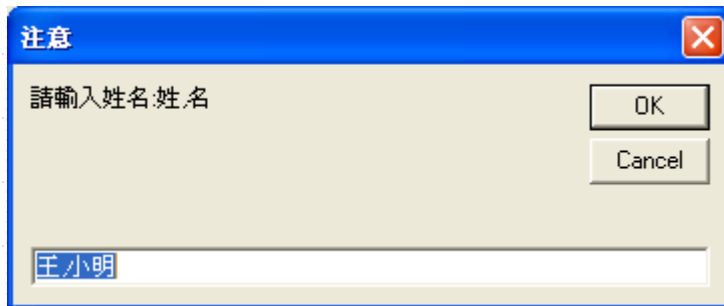
◆ 去除左/右空白字元:LTrim(字串)/RTrim(字串)

■ LTrim(" test ")→"test "

■ RTrim(" test ")→" test"

# 字串函數 (cont.)

- ◆ 字串切割：利用 InputBox 輸入姓名，並且使用字串相關函數將姓與名切割開來，分別存到 `firstname` 與 `lastname` 兩個變數中。



# 字串函數 (cont.)

```
Private Sub Form_Click()  
    Dim name, lastname, firstname As String  
    Dim position As Integer  
    Do  
        name = InputBox("請輸入姓名:姓,名", "注意")  
        name = Trim(name)  
        position = InStr(name, ",")  
    Loop Until (name <> "" And position <> 0)  
    lastname = Left(name, position - 1)  
    firstname = Mid(name, position + 1)  
    Print lastname + "先生,你好"  
End Sub
```

# Try it!

◆ 請撰寫一個程式，將電話號碼分成區碼與號碼兩個部份。

■ 例：(02)12345678

■ 輸出：

◆ 區碼：02

◆ 號碼：12345678



# Visual Basic 程式設計

- ◆ 字串函數
- ◆ 鍵盤事件
- ◆ ActiveX控制項 – MaskedTextBox元件

# 鍵盤與事件

- ◆ Visual Basic提供了許多控制項介面來處理由鍵盤輸入的資料，例如『文字方塊』控制項中的文字通常是由鍵盤所輸入的資料。除此之外，某些特殊按鍵或組合鍵是具有特殊意義的，為了要提供這種服務，Visual Basic提供了三個事件來處理鍵盤操作。

事件名稱	說明
物件_KeyPress	在特定物件上收到由鍵盤按鍵的字元。（僅回應按鍵動作，無法判斷該按鍵是否仍被按住或已放開）
物件_KeyDown	在特定物件上偵測出有按鍵被按住。
物件_KeyUp	在特定物件上偵測出被按住的鍵已放開。

# 鍵盤與事件 (cont.)

當使用者按下鍵盤後，將觸發KeyPress事件與KeyDown事件，放開鍵盤時，則觸發KeyUp事件，不是按下所有的按鈕都會觸發KeyPress事件，只有按下KeyAscii碼的按鍵，才會觸發KeyPress事件。KeyAscii碼如下列所示：

合法的按鍵	傳入的KeyAscii碼值
任何可印出的鍵盤字元	字元的ASCII碼
【 Tab 】 鍵	9
【 Ctrl 】 + 【 A 】 鍵~【 Ctrl 】 + 【 Z 】 鍵	1~26
【 Enter 】 鍵與【 Ctrl 】 + 【 Enter 】 鍵	13 【 Enter 】 鍵與10
【 Backspace 】 鍵與【 Ctrl 】 + 【 Backspace 】 鍵	8 【 Backspace 】 鍵（倒退鍵）與127

# 鍵盤與事件 - KeyPress事件

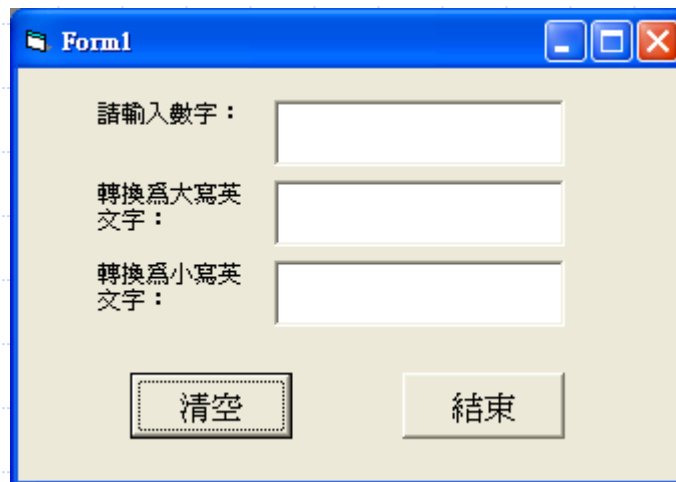
◆ KeyPress事件語法如下：

```
Private Sub物件_KeyPress(KeyAscii As Integer)
    .....
End Sub
```

- **【語法說明】：**
  - **KeyAscii**：傳入代表被按鍵盤的ASCII碼（參閱附錄）。
  - 若在事件程序中更改KeyAscii參數內容，則顯示的字元將跟著被更改。
  - 若將KeyAscii設為0，則會清除剛才輸入的字元。

# 鍵盤與事件 - KeyPress事件 (cont.)

- **【Try it!】**：設計三種自動校正與禁止輸入的文字方塊，如下規定。
  - 第一個文字方塊只允許輸入數字與小數點，輸入其他字元則清除掉所輸入的字元。
  - 第二個文字方塊內，會將小寫字母改成大寫字母，其它字元維持不變。
  - 第三個文字方塊內，會將大寫字母改成小寫字母，其它字元維持不變。



The screenshot shows a Windows application window titled "Form1". It contains three text input fields stacked vertically. The first field is labeled "請輸入數字：" (Please enter a number:). The second field is labeled "轉換為大寫英文字：" (Convert to uppercase English text:). The third field is labeled "轉換為小寫英文字：" (Convert to lowercase English text:). Below the input fields are two buttons: "清空" (Clear) and "結束" (End).

# 鍵盤與事件 -

## KeyDown事件與KeyUp事件

- ◆ **KeyDown與KeyUp事件的觸發時機與動作：**
  - **KeyDown動作：**按下鍵盤按鍵不放時，會觸發KeyDown事件。
  - **KeyUp動作：**放開已按下的按鍵時，會觸發KeyUp事件（恰與KeyDown動作相反）。
  - 可以配合KeyPress事件來檢查按下的是哪一個按鍵。
- ◆ **KeyDown、KeyUp、KeyPress事件提供的功能有些不同：**
  - 若輸入一個字元，則事件發生順序為KeyDown事件、KeyPress事件、KeyUp事件。
  - KeyPress事件可以取得按下字元的ASCII碼，但無法得知鍵盤是否持續被按著，或者是一按下就被放開。
  - KeyUp和KeyDown事件程序可以處理KeyPress事件無法處理的按鍵，例如：功能鍵、編輯鍵與組合鍵。

# 鍵盤與事件 -

## KeyDown事件與KeyUp事件 (cont.)

- ◆ 下列狀況無法觸發KeyDown、KeyUp、KeyPress事件：
  - 當表單出現指令按鈕，並且預設屬性為True時，按下【Enter】鍵，只會觸發指令按鈕的事件，對於其它控制項則沒有作用。
  - 當表單出現的指令按鈕的取消屬性為True時，按【Esc】鍵只會觸發指令按鈕的事件，對於其它控制項則沒有作用。
  - 按【Tab】鍵沒有作用
- ◆ KeyDown與KeyUp事件語法

```
Private Sub物件_KeyDown(KeyCode As Integer , Shift As Integer)  
.....  
End Sub
```

```
Private Sub物件_KeyUp(KeyCode As Integer , Shift As Integer)  
.....  
End Sub
```

# 鍵盤與事件 -

## KeyDown事件與KeyUp事件 (cont.)

### ◆ 【語法說明】：

- 參數KeyCode：KeyCode是鍵盤掃描碼（非ASCII碼）。鍵盤掃描碼如下：

按鍵	掃描碼
A~Z	65~90
0~9	48~57

功能鍵	掃描碼
【F1】 ~ 【F12】	112~123
0~9	48~57

7 Home	8 ↑	9 PgUp
4 ←	5	6 →
1 End	2 ↓	3 PgDn

對應掃描碼



36	38	33
37	12	39
35	40	34



# 鍵盤與事件 -

## KeyDown事件與KeyUp事件 (cont.)

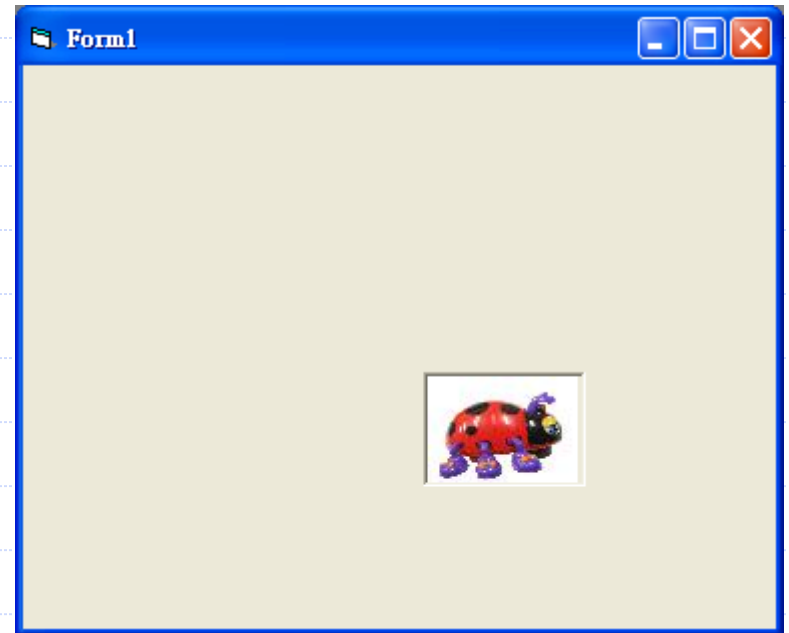
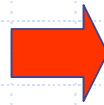
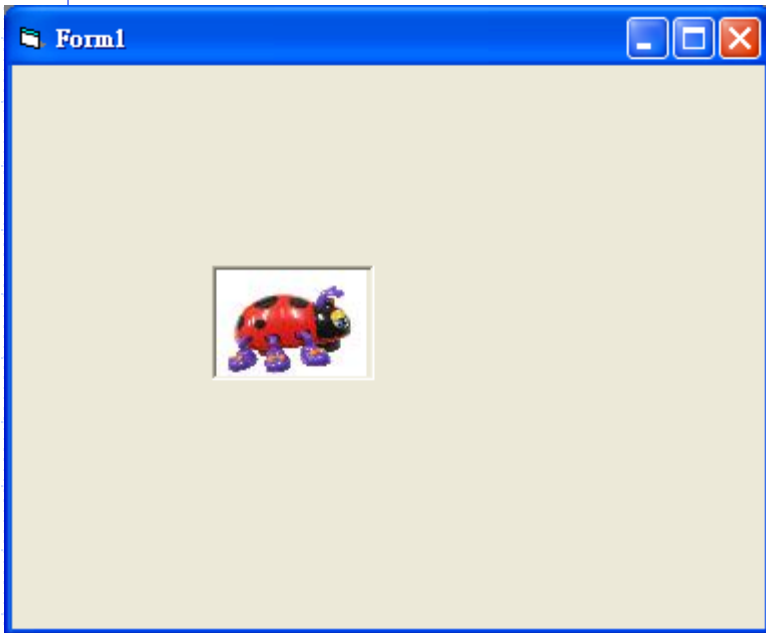
- ◆ 參數Shift：將傳入【Shift】、【Ctrl】、【Alt】等3鍵被按下的情形。如下表：

Shift值	3鍵被按下的形況
0	此3鍵都沒被按下
1	僅【Shift】鍵被按下
2	僅【Ctrl】鍵被按下
3	【Shift】與【Ctrl】鍵同時被按下
4	僅【Alt】鍵被按下
5	【Shift】與【Alt】鍵同時被按下
6	【Ctrl】與【Alt】鍵同時被按下
7	【Shift】、【Ctrl】、【Alt】3鍵同時被按下

# 鍵盤與事件 -

## KeyDown事件與KeyUp事件 (cont.)

- ◆ **【Try it!】**：設計一個同時按下【Ctrl】+【Shift】+【方向】鍵不放可移動圖片的程式，當放開鍵盤時，則會將圖片歸位。
  - 使用物件的.top與.left控制物件的位置。



# Visual Basic 程式設計

- ◆ 字串函數
- ◆ 鍵盤事件
- ◆ **ActiveX控制項 – MaskedTextBox元件**

# ActiveX控制項

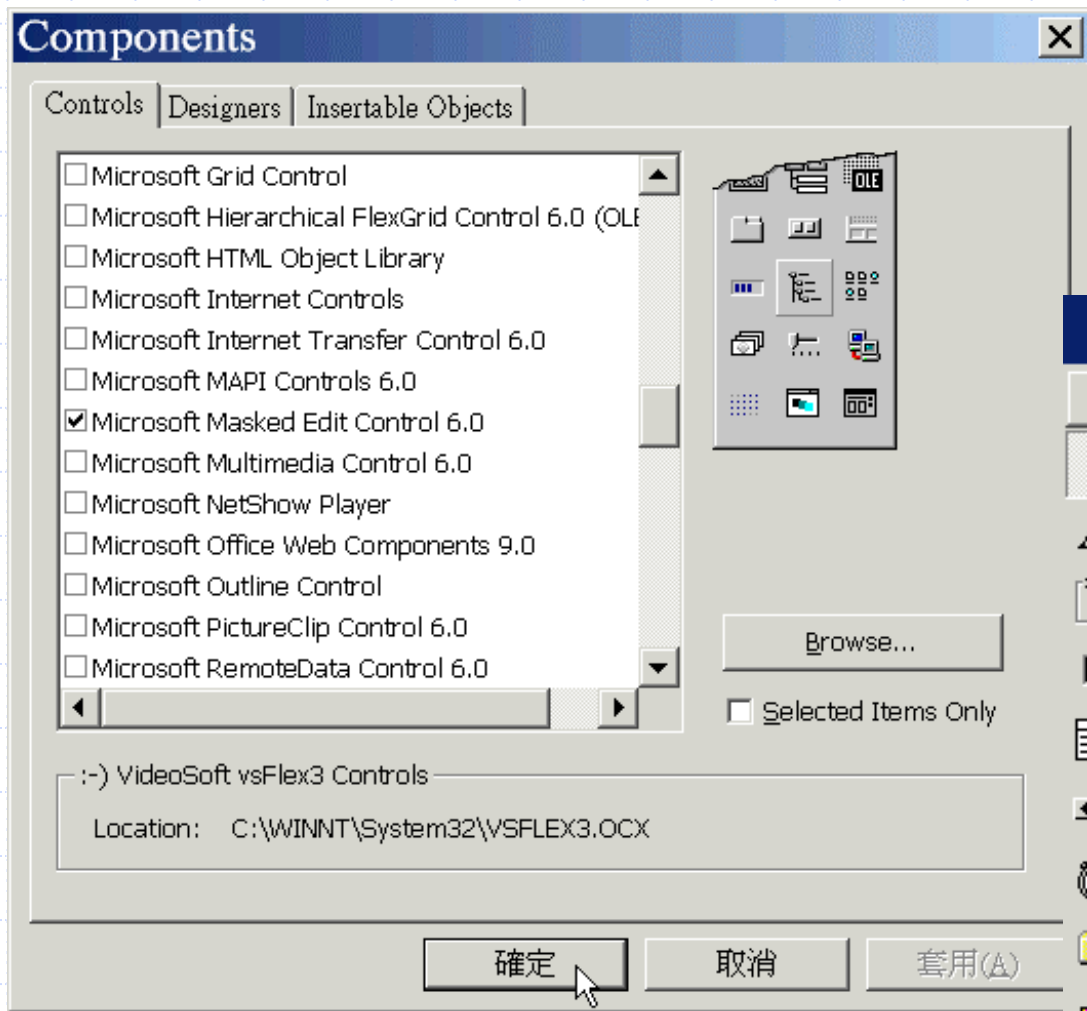
- ◆ **ActiveX**在廣義上是指微軟公司的整個COM架構，但是現在通常用來稱呼基於標準COM介面來實作物件連結與嵌入的ActiveX控制項。
- ◆ 在VB 6.0中，基本控制項為預設會出現，而ActiveX控制項則要手動增加。



按下右鍵

Components...  
Add Tab...

✓ Dockable  
Hide



選擇 Components



新增了 Masked Edit

# MaskEdit ### 遮罩文字方塊

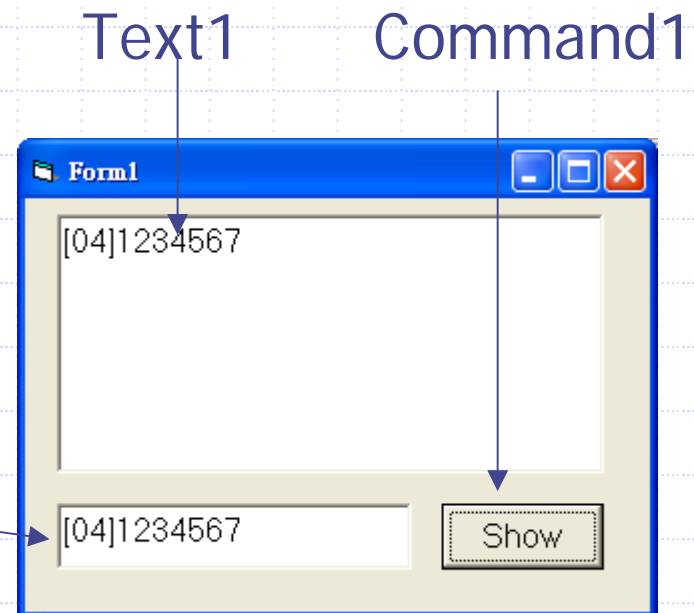
- ◆ 由於TextBox文字方塊無法對輸入的資料做篩選，所以VB 6.0的ActiveX附加元件-MaskEdit，透過Mask屬性的設定，就可按照遮罩篩選樣式防止使用者輸入不合法的字元，而不需編寫任何程式碼就可以完成檢查的工作。因此若要限制使用者輸入特定樣式的字元，或指定輸入字元的個數時，可以使用MaskedTextBox遮罩文字方塊控制項來取代TextBox。

# MaskEdit ### 遮罩文字方塊 (cont.)

## 常用屬性

屬性	
Mask	輸入格式設定
Text	輸入的文字
PromptChar	提示提示字元

MaskedTextBox1



# MaskEdit **##|** 遮罩文字方塊 (cont.)

```
Private Sub Command1_Click()  
    If (Text1.Text = "") Then  
        Text1.Text = MaskedTextBox1.Text  
    Else  
        Text1.Text = Text1.Text + vbNewLine +_  
            MaskedTextBox1.Text  
    End If  
End Sub
```

Mask  
[##]#####



# MaskEdit **##|** 遮罩文字方塊 (cont.)

## ◆ 格式設定字元

#	數字	必要輸入	>	轉大寫	選擇輸入
9	數字	選擇輸入	<	轉小寫	選擇輸入
A	字元	必要輸入	?	字母	必要輸入
a	字元	選擇輸入			

◆ 必要輸入即輸入時不能輸入空格；而選擇輸入則可省略，在省略式，其Text屬性中呈現為空格。

# MaskEdit **##|** 遮罩文字方塊 (cont.)

## ◆ Mask相關範例說明

- (##)#####9
  - ◆ 適用電話號碼。區碼+號碼，一般為8碼，部份地區僅7碼。
- >AA-####
  - ◆ 適用汽車號碼。前兩碼為大寫字母，後四碼為數字，中間用“-”連結。
- #####/##9/##9
  - ◆ 適用日期。如：1987/3/17
- >?#####
  - ◆ 適用身份證字號。第一碼為大寫的英文字母，後九碼為數字。

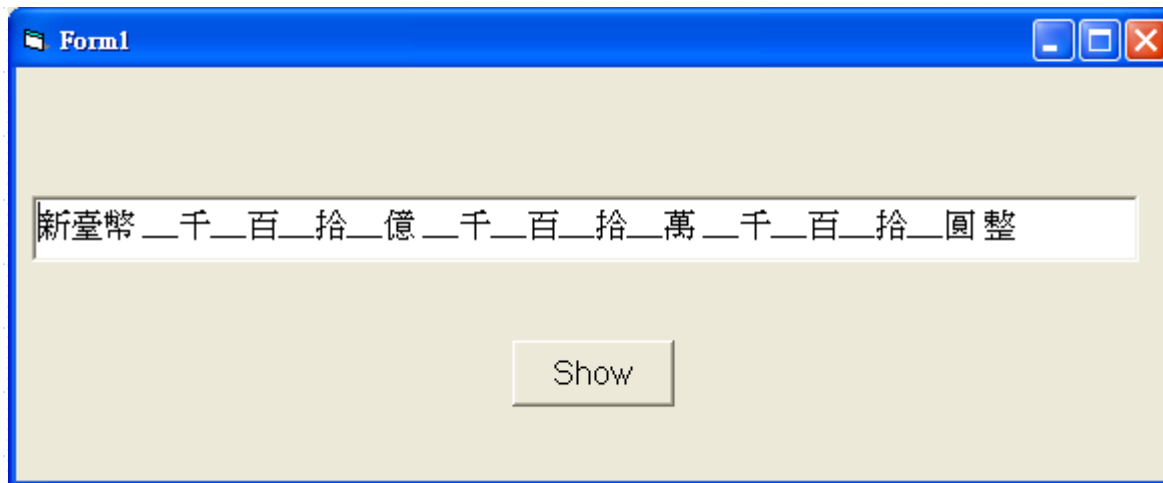
# MaskEdit **##** 遮罩文字方塊 (cont.)

◆ 撰寫一個只能輸入新台幣格式的表單程式。

- 在Mask屬性裡輸入“新臺幣 &&千&&百&&拾&&億&&千&&百&&拾&&萬&&千&&百&&拾&&圓 整”
- “#”與“&”有何不同？
  - ◆ “#”是指定MaskedTextBox只能輸入 0~9，而“&”能輸入任何的可列印字元，因中文佔雙位元組，故指定兩個“&”。

# MaskEdit ### 遮罩文字方塊 (cont.)

- ◆ 讓使用者不論輸入數字0~9或是一~九都顯示壹~玖。



# MaskEdit ### 遮罩文字方塊 (cont.)

```
Private Sub MaskedTextBox1_KeyPress(KeyAscii As Integer)
    Select Case Chr(KeyAscii)
        Case "0", "零", " "
            KeyAscii = Asc("零")
            

請填入相對應的程式碼


        Case "9", "玖"
            KeyAscii = Asc("玖")
        Case Else
            KeyAscii = 0
    End Select
End Sub
```

# 作業 – 身份證驗證

- ◆ 請撰寫一個表單程式，以MaskedTextBox讓使用者輸入身份證字號，並且驗證是否正確。
  - 第一碼英文字代表該組號碼配賦時所屬的縣市〔非出生地〕，
  - 第二碼數字1為男生2為女生，
  - 第3碼至第10碼為任意一串數字，
  - 但含英文字之全部號碼可用公式加以核算該組號碼是否正確，
    - ◆ 例如R123456783， $R=25$ ，
    - ◆ 檢查公式是：
$$2+5*9+1*8+2*7+3*6+4*5+5*4+6*3+7*2+8*1=167$$
，
    - ◆ 其167再除以10求餘數結果。
    - ◆ 其餘數結果的個位數為7以10減去得3(檢查碼)。
    - ◆ 其餘數結果的個位數為0時，則得0(檢查碼)。

# 英文字母對應數字

英文字母	A	B	C	D	E	F	G	H	I	J	K	L	M
對應數字	10	11	12	13	14	15	16	17	34	18	19	20	21
英文字母	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
對應數字	22	35	23	24	25	26	27	28	29	30	41	42	33

# 作業 - 身份證驗證 (cont.)

◆ 執行畫面：

Form1

身份證字號： R123456783

驗證結果： 合法

驗證 結束