

# 程式設計-使用Visual Basic 2010



## 第9章 搜尋與排序

## 循序搜尋(ch9\循序搜尋\)




- 找出成績陣列中是否包含成績為**59**分的學生，找到一位即可。
- 想一想
  - 想要找出班上第一次段考成績是否有**59**分的學生，可能需要有全班成績單，從頭開始依序找，
  - (1)第一位學生成績是否為**59**分，若是，則輸出「找到**59**分的學生」程式結束；
  - (2)否則，比較第二位學生的成績是否為**59**分，若是，則輸出「找到**59**分的學生」程式結束；
  - (3)依此類推，直到比較到最後一位學生的成績為止。
  - (4)若全部都找不到成績**59**分的學生，則輸出：「找不到**59**分的學生」。

# 循序搜尋(ch9\循序搜尋\)



- 舉例說明

- 第一位學生的成績(60)是否為59分，若是，則輸出「找到59分的學生」，程式結束。



60	90	44	98	50
----	----	----	----	----

- 否則，比較第二位學生的成績(90)是否為59分，若是，則輸出「找到59分的學生」，程式結束。



60	90	44	98	50
----	----	----	----	----

## 循序搜尋(ch9\循序搜尋\)



- 否則，比較第三位學生的成績(44)是否為59分，若是，則輸出「找到59分的學生」，程式結束。

60	90	44	98	50
----	----	----	----	----

- 否則，比較第四位學生的成績(98)是否為59分，若是，則輸出「找到59分的學生」，程式結束。

60	90	44	98	50
----	----	----	----	----

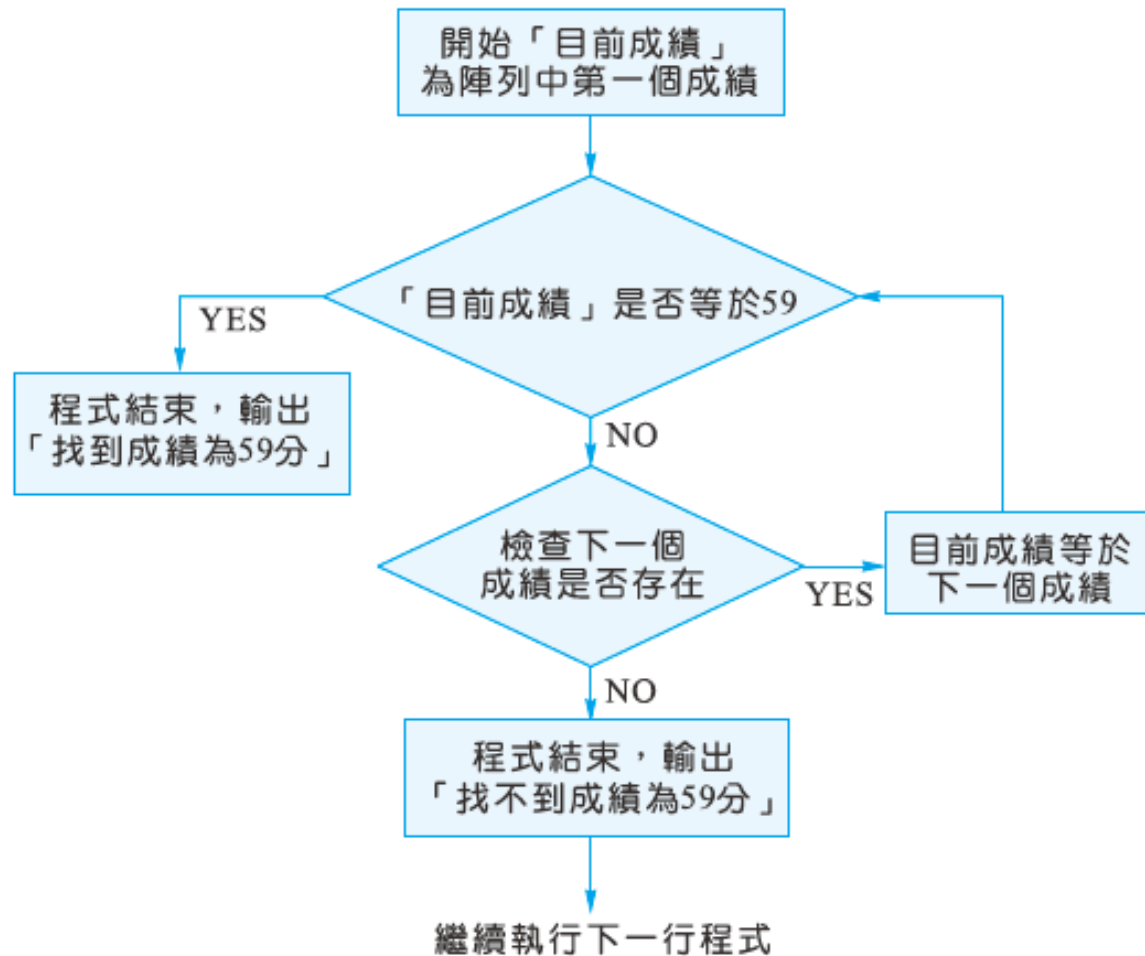
- 否則，比較第五位學生的成績(50)是否為59分，若是，則輸出「找到59分的學生」，程式結束。

60	90	44	98	50
----	----	----	----	----

- 若全部都找不到成績59分的學生，則輸出「找不到59分的學生」。

# 循序搜尋(ch9\循序搜尋\)

- 流程圖表示



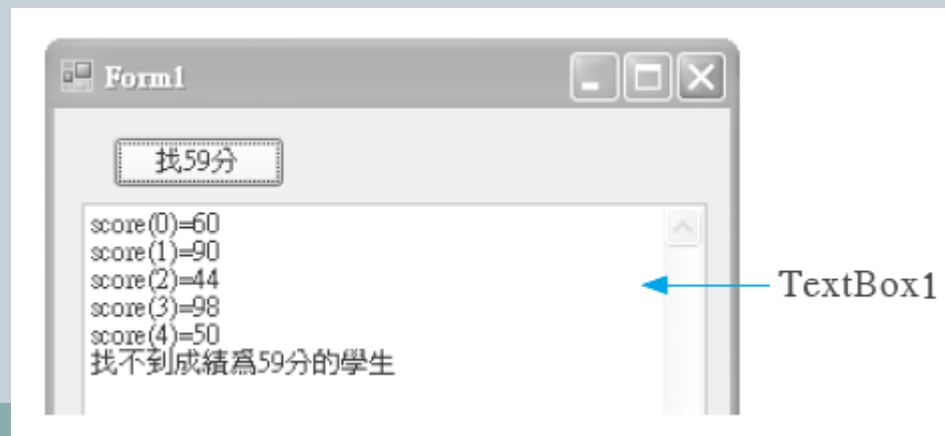
# 循序搜尋(ch9\循序搜尋\)

- 演算法說明

- 這樣的演算法需要成績陣列，一個迴圈(**For**)用於檢查成績陣列的每一個元素，若找到一個元素等於**59**分，則輸出「找到**59**分的學生」；若全部都找不到成績**59**分的學生，則輸出「找不到**59**分的學生」。

- 預覽結果

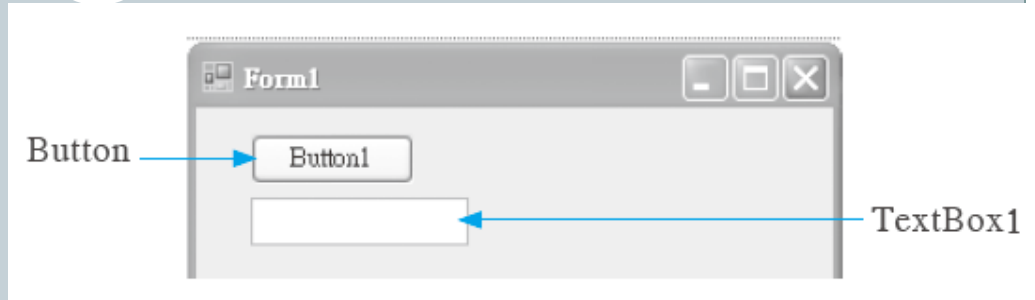
- 按下「找59分」按鈕，結果顯示在**TextBox1**中。



# 循序搜尋(ch9\循序搜尋\)

- 表單配置

在表單中新增一個  
**TextBox**與一個**Button**。



- 屬性設定

物件	屬性	設定值
Button1	Text	找59分
TextBox1	ScrollBars	Both
TextBox1	Multiline	True
TextBox1	Size	270,215



## 循序搜尋(ch9\循序搜尋\)

### ● 程式碼新增與解說

- 點選「找59分」按鈕，新增程式碼在Button1\_Click函式中，如下方程式區塊。

行號	程式碼
3	Dim score() = {60, 90, 44, 98, 50}
4	Dim found As Integer = 0
5	For i = 0 To 4
6	TextBox1.Text = TextBox1.Text & "score(" & i & ")=" & score(i) & vbNewLine
7	If (score(i) = 59) Then
8	found = 1
9	TextBox1.Text = TextBox1.Text & "找到成績為59分的學生" & vbNewLine
10	End If
11	Next
12	If (found = 0) Then
13	TextBox1.Text = TextBox1.Text & "找不到成績為59分的學生"
14	End If



## 二元搜尋(ch9\二元搜尋\)



- 找出已排序成績陣列中是否包含成績為**59**分的學生。
- 想一想
- 第**9-1-1**節以循序搜尋方式找尋資料，從頭到尾依序找尋，但對已經由小到大排序好的資料可以使用二分搜尋方式加快找尋速度，因為已經排序可以從中間開始找
  - 若要找的元素比中間元素值大，則往右邊找；
  - 若要找的元素比中間元素值小，則往左邊找，依此類推，直到找到為止。

# 二元搜尋(ch9\二元搜尋\)



## ● 舉例說明

- 假設已排序的十位學生的成績陣列，如下圖，以二分搜尋方式找尋成績為**59**分的學生。

45	59	62	67	70	78	83	85	88	92
----	----	----	----	----	----	----	----	----	----

- 取第一位到第十位學生成績中間的那位學生，第六位學生的成績**(78)** 是否為**59**分，若是，則輸出「找到**59**分的學生」程式結束。


45	59	62	67	70	78	83	85	88	92
----	----	----	----	----	----	----	----	----	----



## 二元搜尋(ch9\二元搜尋\)




- 否則，因為**59**分小於**78**分，往左邊取出由第一位到第五位學生成績中間的那位學生，第三位學生的成績(**62**)是否為**59**分，若是，則輸出「找到**59**分的學生」程式結束。



45	59	62	67	70	78	83	85	88	92
----	----	----	----	----	----	----	----	----	----

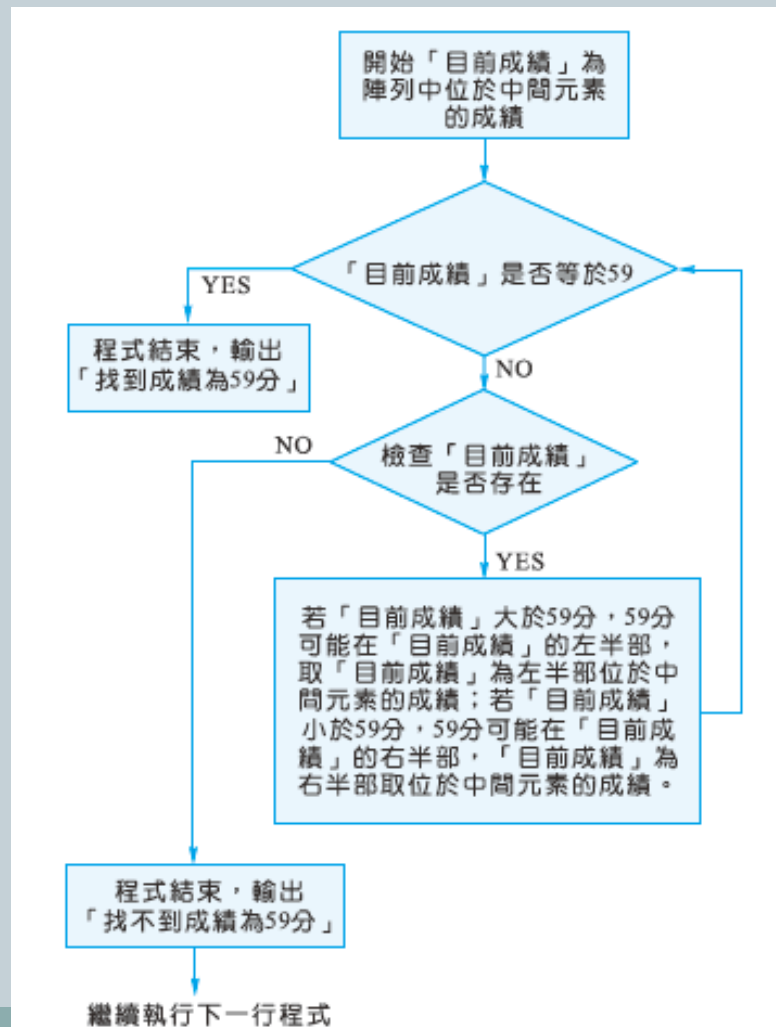
- 否則，因為**59**分小於**62**分，往左邊取出由第一位到第二位學生成績中間的那位學生，第二位學生的成績(**59**)是否為**59**分，找到**59**分的學生，輸出「找到**59**分的學生」程式結束。



45	59	62	67	70	78	83	85	88	92
----	----	----	----	----	----	----	----	----	----

# 二元搜尋 (ch9\二元搜尋\)

- 流程圖表示



# 二元搜尋(ch9\二元搜尋\)



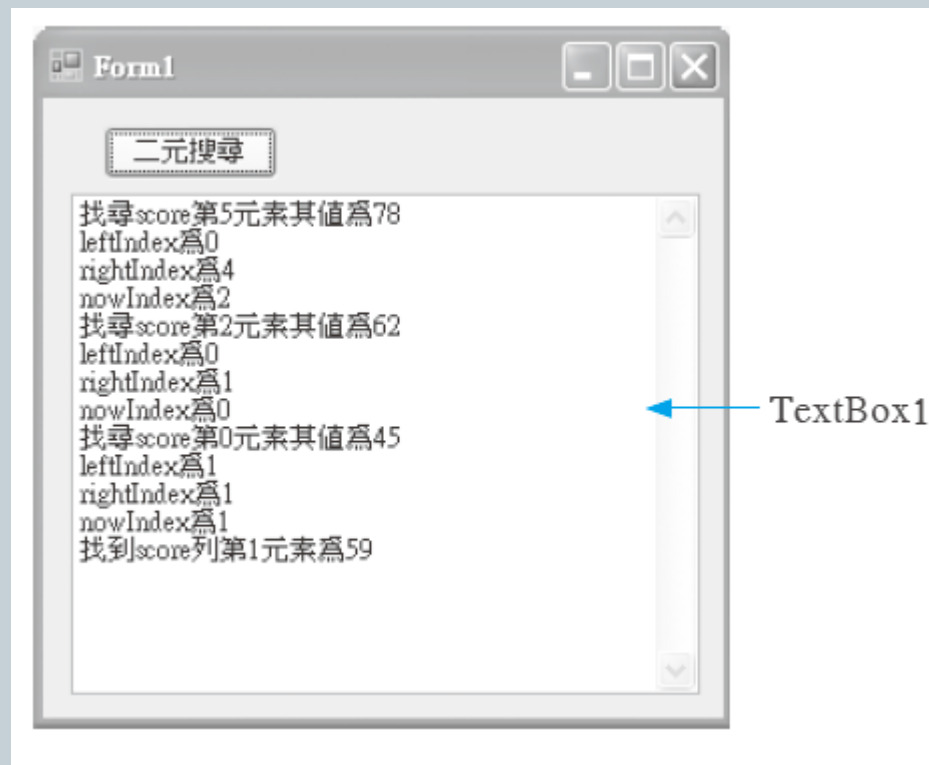
- 演算法說明

- 這樣的演算法需要一個成績陣列，事先將成績陣列由小到大排序好，再利用一個迴圈(**while**)用於檢查「目前成績」是否等於**59**分，若找到一個成績等於**59**分，則輸出「找到**59**分的學生」
- 否則，若「目前成績」大於**59**分，**59**分可能在「目前成績」的左半部，「目前成績」為左半部陣列元素取位於中間元素的成績
- 若「目前成績」小於**59**分，**59**分可能在「目前成績」的右半部，「目前成績」為右半部成績陣列取位於中間元素的成績。
- 若找不到可以比較的「目前成績」，則輸出「找不到**59**分的學生」。

# 二元搜尋(ch9\二元搜尋\)

- 預覽結果

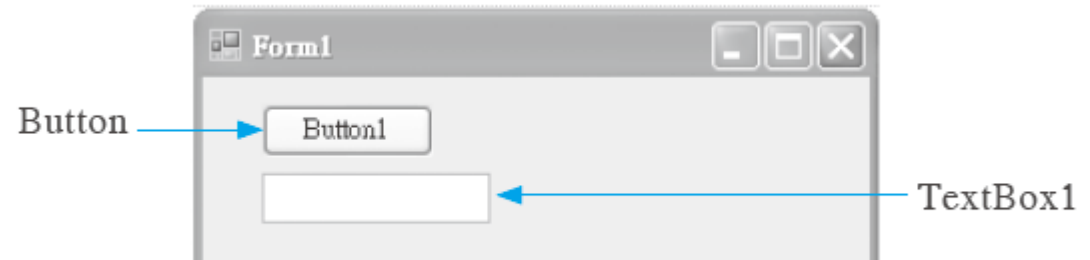
- 按下「二元搜尋」按鈕，結果顯示在**TextBox1**中。



# 二元搜尋(ch9\二元搜尋\)

- 表單配置

在表單中新增一個  
**TextBox**與一個**Button**。



- 屬性設定

物件	屬性	設定值
Button1	Text	二元搜尋
TextBox1	ScrollBars	Both
TextBox1	Multiline	True
TextBox1	Size	270,215



## 二元搜尋(ch9\二元搜尋\)

- 程式碼新增與解說

- 點選「二元搜尋」按鈕，新增程式碼在Button1\_Click函式中，如下方程式區塊。

行號	程式碼
3	Dim score() As Integer = {45, 59, 62, 67, 70, 78, 83, 85, 88, 92}
4	Dim nowIndex As Integer = 5
5	Dim leftIndex As Integer = 0
6	Dim rightIndex As Integer = 9
7	Do While score(nowIndex) <> 59
8	TextBox1.Text = TextBox1.Text & "找尋陣列score第" & nowIndex & "元素其值為" & score(nowIndex) & vbNewLine
9	If leftIndex = rightIndex Then
10	Exit Do
11	End If
12	If score(nowIndex) > 59 Then
13	rightIndex = nowIndex - 1
14	Else
15	leftIndex = nowIndex + 1
16	End If
17	nowIndex = (leftIndex + rightIndex) / 2
18	TextBox1.Text = TextBox1.Text & "leftIndex為" & leftIndex & vbNewLine
19	TextBox1.Text = TextBox1.Text & "rightIndex為" & rightIndex & vbNewLine
20	TextBox1.Text = TextBox1.Text & "nowIndex為" & nowIndex & vbNewLine
21	Loop



# 二元搜尋(ch9\二元搜尋\)



行號	程式碼
22	If score(nowIndex) = 59 Then
23	TextBox1.Text = TextBox1.Text & "找到陣列score第" & nowIndex & "元素為59" & vbNewLine
24	Else
25	TextBox1.Text = TextBox1.Text & "找不到成績為59分的資料"
26	End If

# 二元搜尋(ch9\二元搜尋\)



## ● 解說

- 第3行：宣告整數陣列**score**，初始化為**10**個元素的陣列，從第**1**個到第**10**個元素分別是「**45, 59, 62, 67, 70, 78, 83, 85, 88, 92**」。
- 第4行：宣告**nowIndex**為整數變數，且初始化為**5**，**score(nowIndex)**指向「目前成績」。
- 第5行：宣告**leftIndex**為整數變數，且初始化為**0**，決定搜尋範圍的左邊界。
- 第6行：宣告**rightIndex**為整數變數，且初始化為**9**，決定搜尋範圍的右邊界。
- 第7到21行：**Do while...Loop**迴圈中使用變數**nowIndex**，**nowIndex**為「目前成績」的陣列索引，判斷**score(nowIndex)**是否為**59**，若不是，則繼續迴圈；若是，則跳出迴圈。

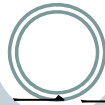
## 二元搜尋(ch9\二元搜尋\)



- 解說

- 第8行：顯示目前成績score(nowIndex)於TextBox1。
- 第9到11行：若leftIndex與rightIndex相等，表示搜尋範圍已經沒有元素了。
- 第12到16行：若「目前成績」大於59，表示搜尋左半部，將right Index改成nowIndex-1(第13行)；否則，表示搜尋右半部，將leftIndex改成nowIndex+1(第15行)，讓陣列索引變數leftIndex與rightIndex指向新的搜尋範圍。
- 第17行：nowIndex改成取變數leftIndex與rightIndex指向新的搜尋範圍的中間。
- 第18到20行：顯示leftIndex與rightIndex與nowIndex於TextBox1。
- 第22到26行：若score(nowIndex)等於59，表示找到成績為59分的資料；否則，表示找不到成績為59分的資料。

# 氣泡排序(ch9\氣泡排序\)



- 隨機產生一個陣列的五個元素，並將這五個元素由小到大排序。
- 想一想
  - ✧ 現在有一簡單的想法：若要將五個元素中最大的元素放到陣列最後一個元素的位置，
  - ✧ 可以比較第一個元素與第二個元素，若第一個元素比第二個元素大，則第一個元素與第二個元素交換；
  - ✧ 再比較第二個元素與第三個元素，若第二個元素比第三個元素大，則第二個元素與第三個元素交換；目前第三個元素為三個元素中最大的，
  - ✧ 再比較第三個元素與第四個元素，若第三個元素比第四個元素大，則第三個元素與第四個元素交換；目前第四個元素為四個元素中最大的，
  - ✧ 再比較第四個元素與第五個元素，若第四個元素比第五個元素大，則第四個元素與第五個元素交換，目前第五個元素為五個元素最大的，如此我們已經將最大元素放到陣列最後一個元素。

# 氣泡排序(ch9\氣泡排序\)



- ✧ 依此類推，將範圍改成第一到第四個元素，照上述方式可以將四個元素中最大的元素放到陣列第四個元素。
- ✧ 同理，將範圍改成第一到第三個元素，照上述方式可以將三個元素中最大元素放到陣列第三個元素。
- ✧ 依此類推，將範圍改成第一到第二個元素，照上述方式可以將兩個元素中最大元素放到陣列第二個元素。
- ✧ 到此已經完成陣列資料的排序，此排序方法為「**氣泡排序**」。

# 氣泡排序(ch9\氣泡排序\)

- 舉例說明

- 假設隨機產生五個陣列元素，如下圖。

60	90	44	82	50
----	----	----	----	----

- 比較第一個元素(**60**)與第二個(**90**)元素，若第一個元素比第二個元素大，則第一個元素與第二個元素交換。

60	90	44	82	50

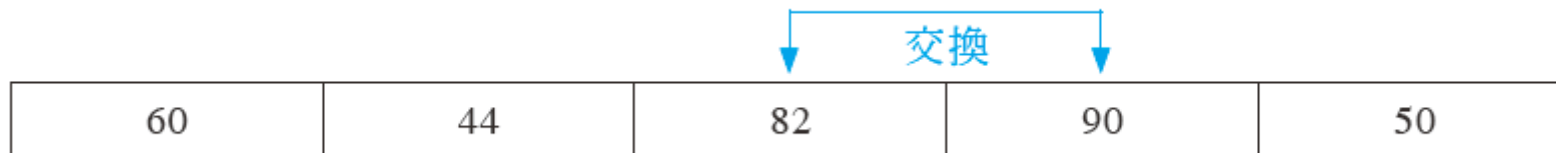
- 再比較第二個元素(**90**)與第三個元素(**44**)，若第二個元素比第三個元素大，則第二個元素與第三個元素交換。目前第三個元素(**90**)為三個元素中最大的。

60	44	90	82	50

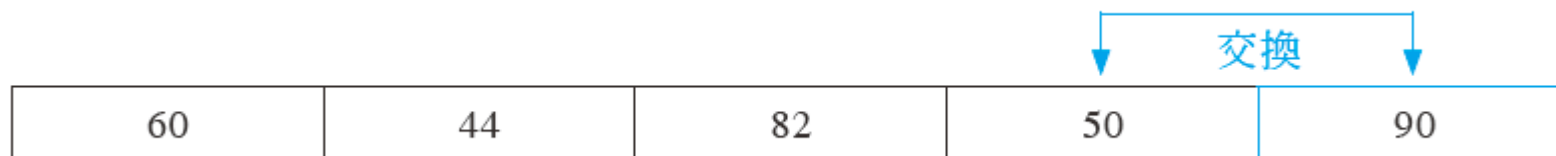


## 氣泡排序(ch9\氣泡排序\)

- 再比較第三個元素(**90**)與第四個元素(**82**)，若第三個元素比第四個元素大，則第三個元素與第四個元素交換。目前第四個元素(**90**)為四個元素中最大的。



- 再比較第四個元素(**90**)與第五個元素(**50**)，若第四個元素比第五個元素大，則第四個元素與第五個元素交換，目前第五個元素(**90**)為五個元素中最大的。如此我們已經將最大元素放到陣列最後一個元素。

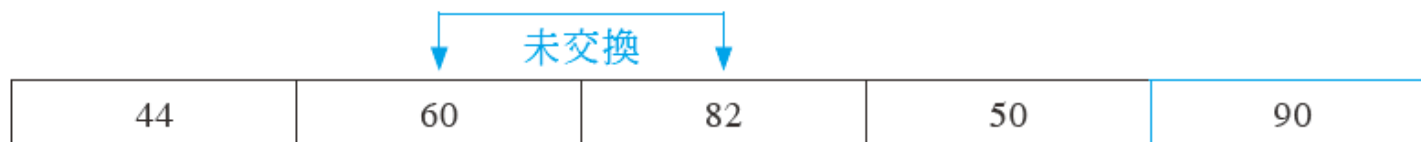


## 氣泡排序(ch9\氣泡排序\)

- 依此類推，將範圍改成第一到第四個元素，比較第一個元素(**60**)與第二個(**44**)元素，若第一個元素比第二個元素大，則第一個元素與第二個元素交換。



- 比較第二個元素(**60**)與第三個(**82**)元素，若第二個元素比第三個元素大，則第二個元素與第三個元素交換。

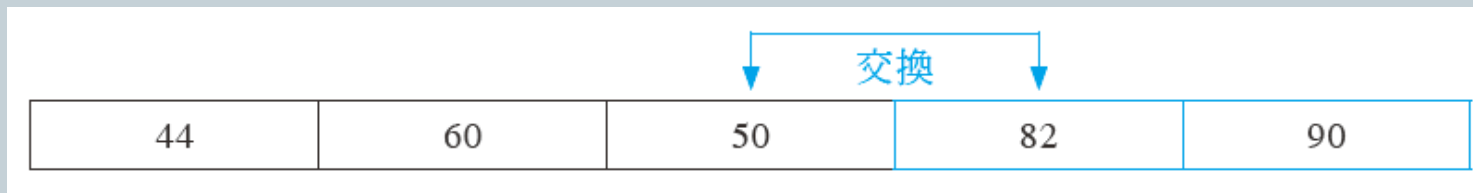




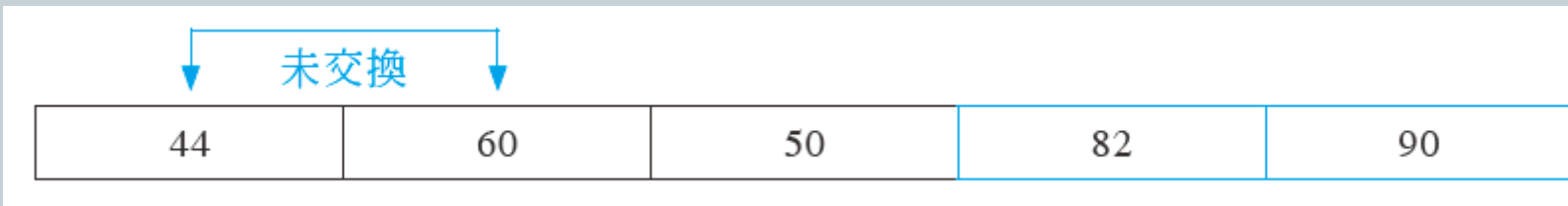
## 氣泡排序(ch9\氣泡排序\)



- 比較第三個元素(**82**)與第四個(**50**)元素，若第三個元素比第四個元素大，則第三個元素與第四個元素交換。照上述方式，可以將四個元素中最大的元素放到陣列第四個元素。

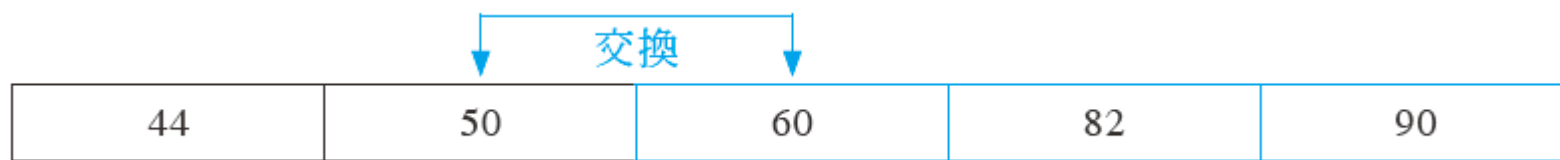


- 依此類推，將範圍改成第一到第三個元素，比較第一個元素(**44**)與第二個(**60**)元素，若第一個元素比第二個元素大，則第一個元素與第二個元素交換。

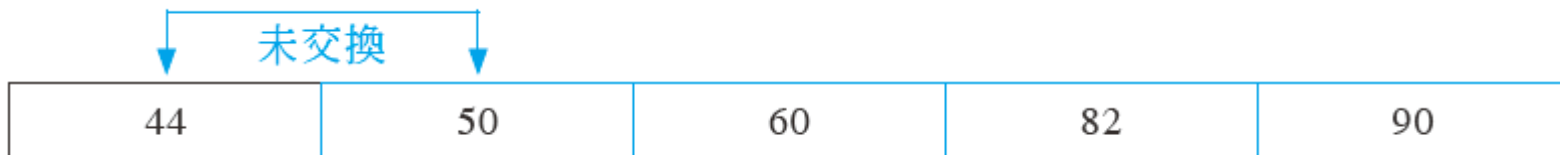


## 氣泡排序(ch9\氣泡排序\)

- 比較第二個元素(**60**)與第三個(**50**)元素，若第二個元素比第三個元素大，則第二個元素與第三個元素交換。照上述方式，可以將三個元素中最大的元素放到陣列第三個元素。

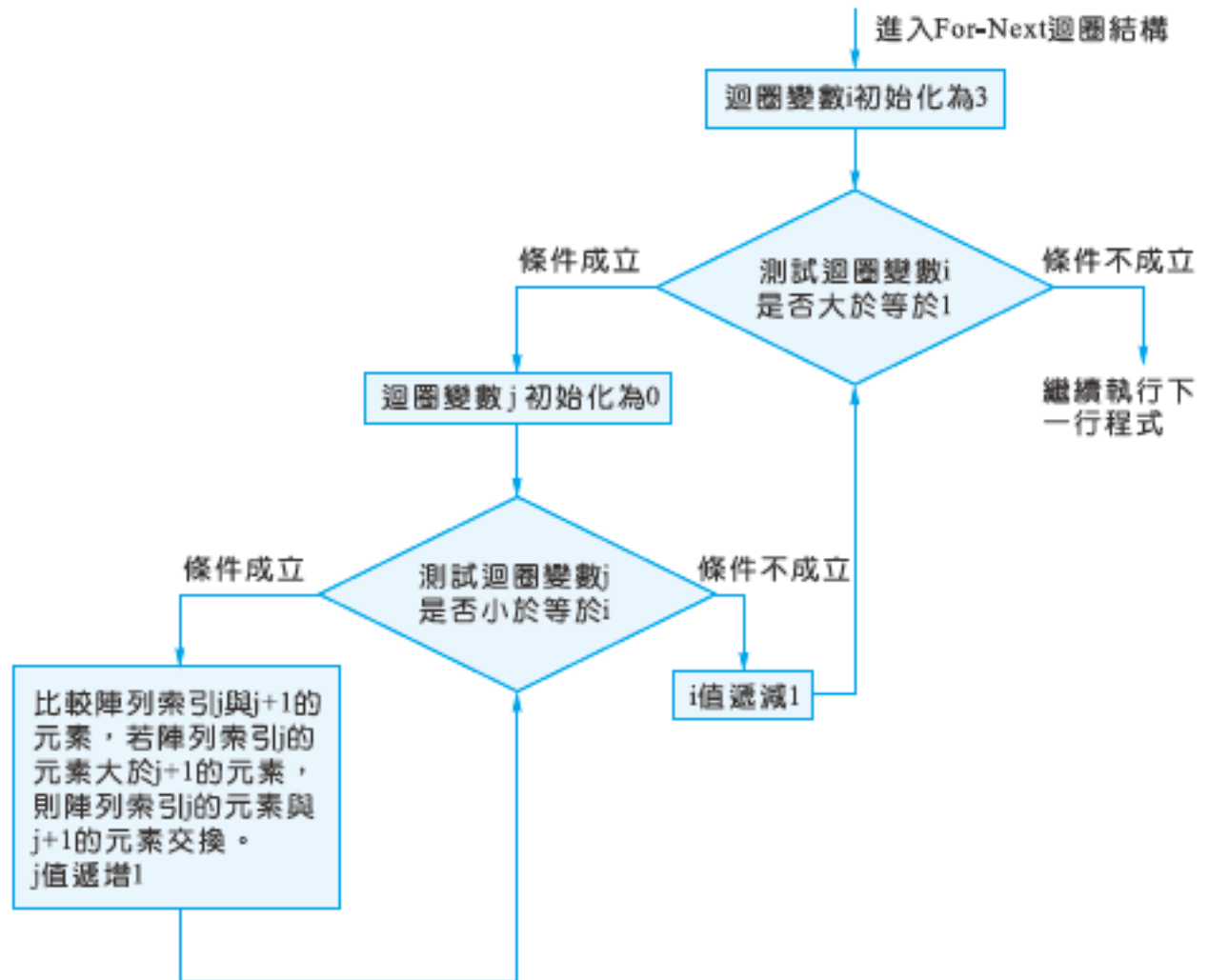


- 依此類推，將範圍改成第一到第二個元素，比較第一個元素(**44**)與第二個(**60**)元素，若第一個元素比第二個元素大，則第一個元素與第二個元素交換。照上述方式可以將兩個元素中最大的元素放到陣列第二個元素，範圍為一個元素就不用排序了，到此已完成排序。



# 氣泡排序(ch9\氣泡排序\)

- 流程圖表示



# 氣泡排序(ch9\氣泡排序\)



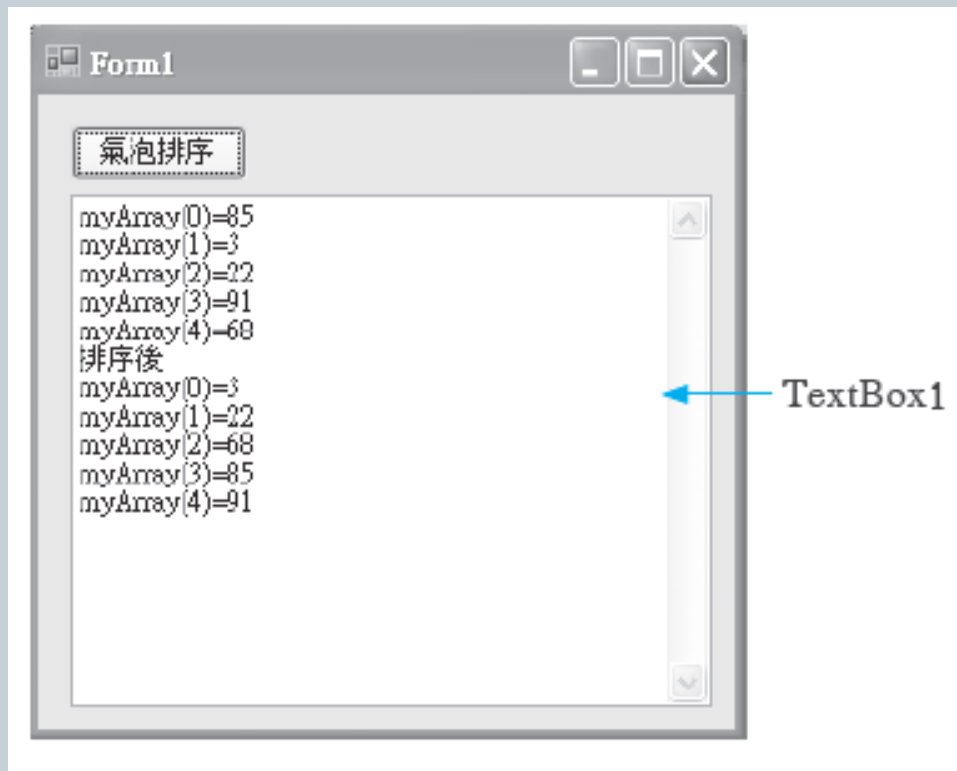
## ● 演算法說明

- 本範例需要一個陣列，隨機產生五個數置於陣列中，使用巢狀迴圈(**For**)，外層迴圈控制要排序陣列元素的上限範圍；內層迴圈用於從頭比較相鄰兩元素。
- 在內層迴圈中，前面比後面大則交換；再比較下兩個相鄰的元素，直到外層迴圈所限制的上限，可以確保最大元素在最後面。
- 內層迴圈結束，外層迴圈變數減一，縮小排序範圍，內層迴圈從頭比較相鄰兩元素，前面比後面大則交換；再比較下兩個相鄰的元素，直到外層迴圈所限制的上限，可以確保縮小排序範圍中最大元素在最後面。
- 重複上述步驟，縮小排序範圍，將最大置於最後面，直到剩下一個元素，就排序完成。

# 氣泡排序(ch9\氣泡排序\)

- 預覽結果

- 按下「氣泡排序」按鈕，結果顯示在**TextBox1**中。

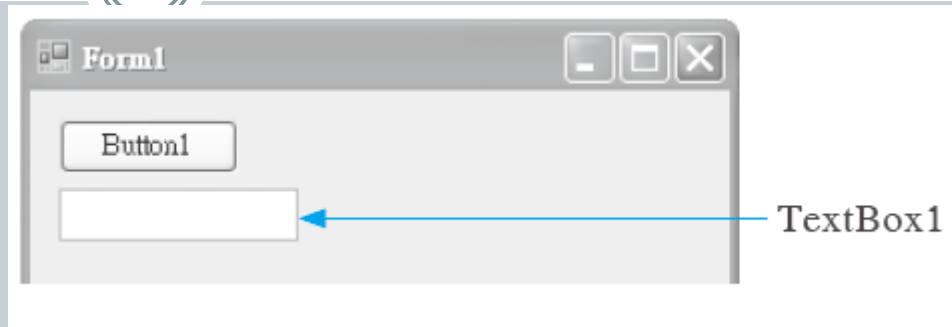


# 氣泡排序(ch9\氣泡排序\)

- 表單配置

在表單中新增一個

**TextBox**與一個**Button**。



- 屬性設定

物件	屬性	設定值
Button1	Text	氣泡排序
TextBox1	ScrollBars	Both
TextBox1	Multiline	True
TextBox1	Size	270,215



## 氣泡排序(ch9\氣泡排序\)

- 程式碼新增與解說

- 點選「氣泡排序」按鈕，新增程式碼在**Button1\_Click**函式中，如下方程式區塊。

行號	程式碼
3	Dim myArray(5) As Integer
4	Dim temp As Integer
5	Randomize()
6	For i As Integer = 0 To 4
7	myArray(i) = CInt(Int(Rnd() * 100))
8	TextBox1.Text = TextBox1.Text & "myArray(" & i & ")=" & myArray(i) & vbNewLine
9	Next
10	For i As Integer = 3 To 1 Step -1
11	For j As Integer = 0 To i
12	If myArray(j) > myArray(j + 1) Then
13	temp = myArray(j + 1)
14	myArray(j + 1) = myArray(j)
15	myArray(j) = temp
16	End If
17	Next
18	Next
19	TextBox1.Text = TextBox1.Text & "排序後" & vbNewLine
20	For i As Integer = 0 To 4
21	TextBox1.Text = TextBox1.Text & "myArray(" & i & ")=" & myArray(i) & vbNewLine
22	Next

# 氣泡排序(ch9\氣泡排序\)



- 解說

- 第3行：宣告五個元素的整數陣列**myArray**。
- 第4行：宣告**temp**為整數變數。
- 第5行：初始化隨機函式。
- 第6到9行：使用**For**迴圈隨機產生陣列**myArray**元素的值，其值為介於0到99的整數。顯示陣列**myArray**元素值於**TextBox1**。
- 第10到18行：氣泡排序演算法，外層迴圈變數**i**，控制內層迴圈變數**j**的上限，迴圈變數**i**由3到1，每次遞減1；內層迴圈**j**由0到**i**，每次遞增1。第12到16行比較相鄰兩數，前面比後面大就交換，第13到15行表示交換兩數
- 第19行：顯示「排序後」於**TextBox1**。
- 第20到22行：顯示排序後陣列所有元素。

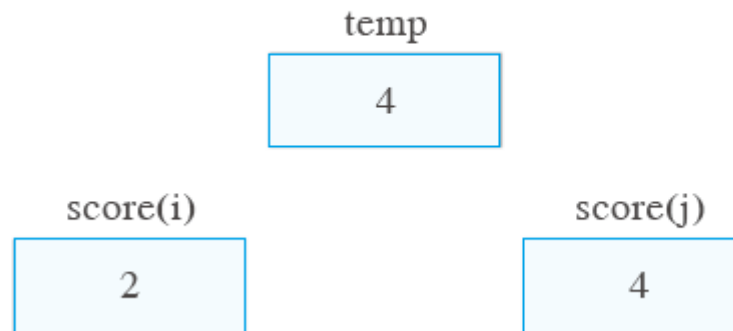


# 氣泡排序(ch9\氣泡排序\)

- 交換兩數

- 程式中交換兩數要使用暫存變數，要先將兩數其中一數暫存於暫存變數，如此交換過程之中才不會有資料遺失問題，假設暫存變數令為**temp**，解說如下，假設交換陣列**score(i)**與**score(j)**兩元素。

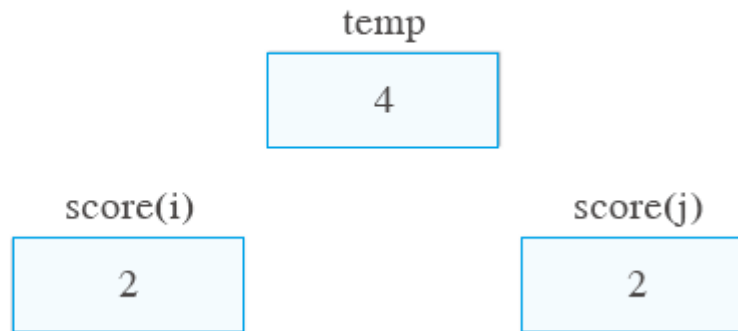
執行temp=score(j)後記憶體狀態。



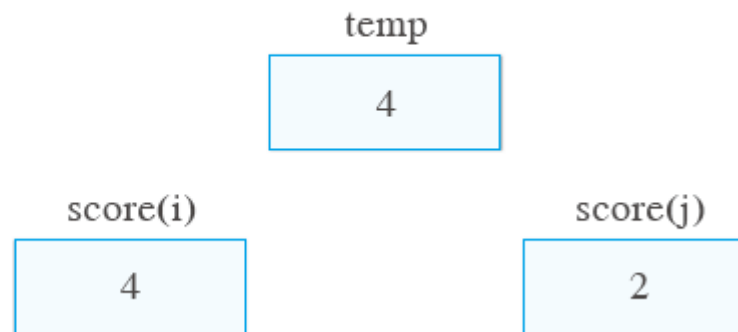
# 氣泡排序(ch9\氣泡排序\)



執行 $\text{score}(j)=\text{score}(i)$ 後記憶體狀態。



執行 $\text{score}(i)=\text{temp}$ 後記憶體狀態。



# 選擇排序前言 - 找出陣列中最大值

(  ch9\找出陣列中最大值\)

- 找出成績陣列中的最高分，假設陣列中只有**5**個元素，且每個元素的值皆不相同。
- 想一想
  - 想要找出班上第一次段考成績最高分的學生，可能需要有全班成績，從頭開始依序找：**(1)**先令「目前為止最高分的成績」為第一個學生成績。**(2)**比較「目前為止最高分的成績」與第二個學生的成績，較高者暫存到「目前為止最高分的成績」。**(3)**再取「目前為止最高分的成績」與第三個學生的成績相比較，較高者暫存到「目前為止最高分的成績」。**(4)**再比較「目前為止最高分的成績」與第四個學生的成績，較高者暫存到「目前為止最高分的成績」。**(5)**依此類推，直到比較到最後一個學生的成績為止，如此最高分就會被找到。

# 選擇排序前言 - 找出陣列中最大值

## (ch9\找出陣列中最大值\)

### ● 舉例說明

- 1. 令「目前為止最高分的成績」為第一個學生的成績(60)。

60	←目前為止最高分			
60	90	44	98	50

- 2. 比較第一個(60)跟第二個學生(90)的成績，較高者(90)暫存到「目前為止最高分的成績」。

90	←目前為止最高分			
↓				
60	90	44	98	50

- 3. 「目前為止最高分的成績」(90)與第三個學生(44)的成績比較，「目前為止最高分的成績」不變。

90	←目前為止最高分			
↓				
60	90	44	98	50

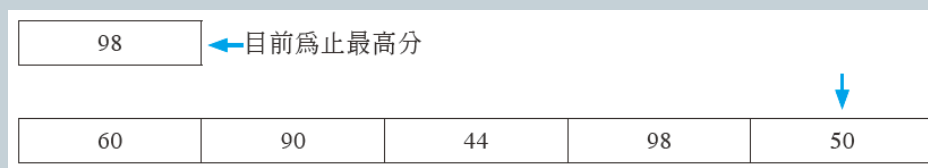
# 選擇排序前言 - 找出陣列中最大值

## ( ch9\找出陣列中最大值\)

- 4. 「目前為止最高分的成績」(90)與第四個學生(98)的成績比較，較高者(98)暫存到「目前為止最高分的成績」。



- 5. 「目前為止最高分的成績」(98)與第五個學生(50)的成績比較，「目前為止最高分的成績」不變。

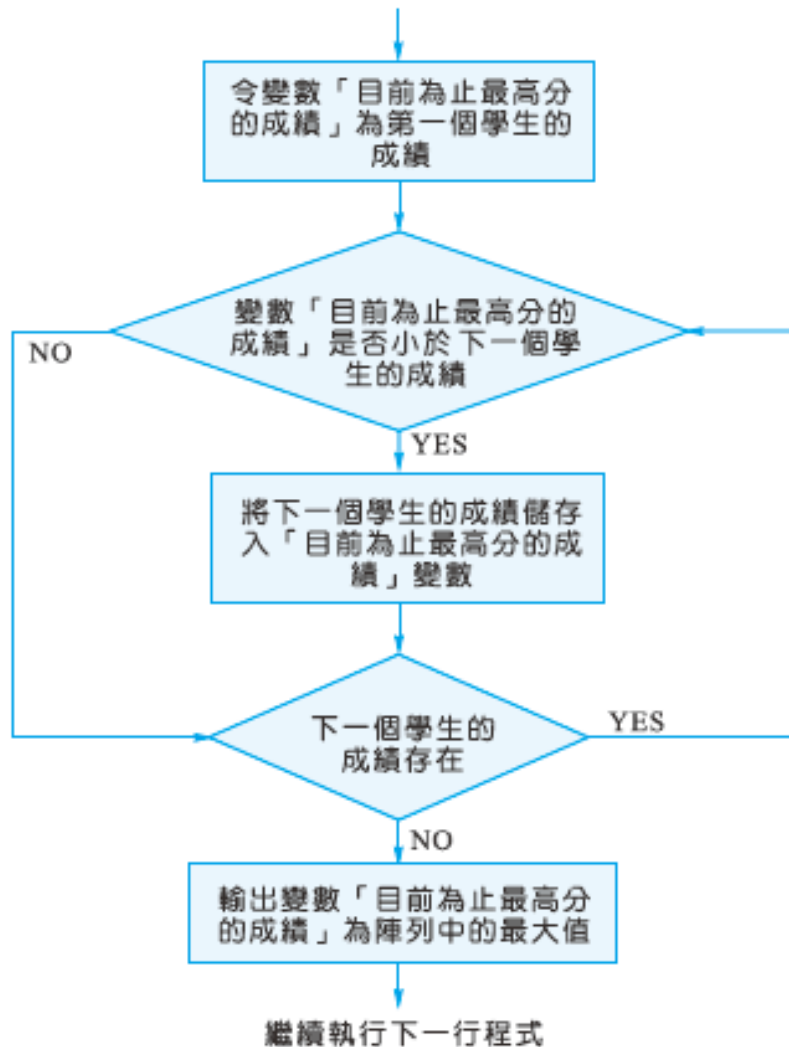


- 6. 比較結束，結果已存入「目前為止最高分的成績」(98)。

# 選擇排序前言 - 找出陣列中最大值

## ( ch9\找出陣列中最大值\)

- 流程圖表示



# 選擇排序前言 - 找出陣列中最大值

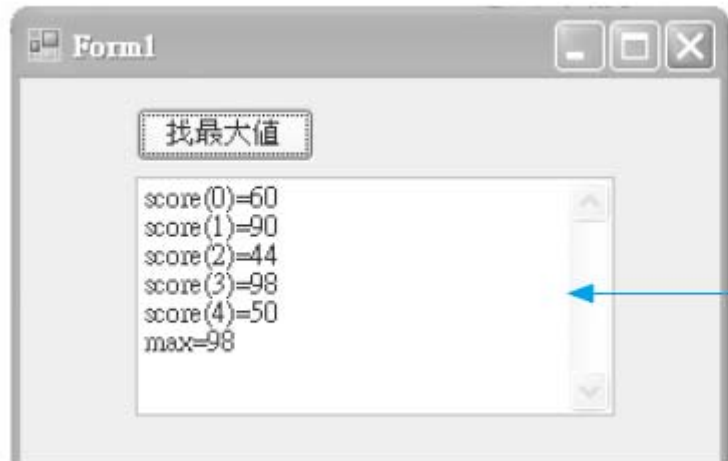
(  ch9\找出陣列中最大值\)

## ● 演算說明

- 這樣的演算法需要一個全班成績陣列，以及一個迴圈(**For**)用於檢查全班成績陣列的每一個元素；一個變數用於儲存「目前為止最高分的成績」；一個選擇判斷(**IF**)用於比較迴圈目前所檢查的學生成績與「目前為止最高分的成績」變數，若「目前為止最高分的成績」變數小於迴圈目前所檢查的學生成績，則「目前為止最高分的成績」變數設定為迴圈目前所檢查的學生成績。

## ● 預覽結果

按下「找最大值」按鈕，結果顯示在  
**TextBox1**中。



Form1

找最大值

score(0)=60  
score(1)=90  
score(2)=44  
score(3)=98  
score(4)=50  
max=98

TextBox1

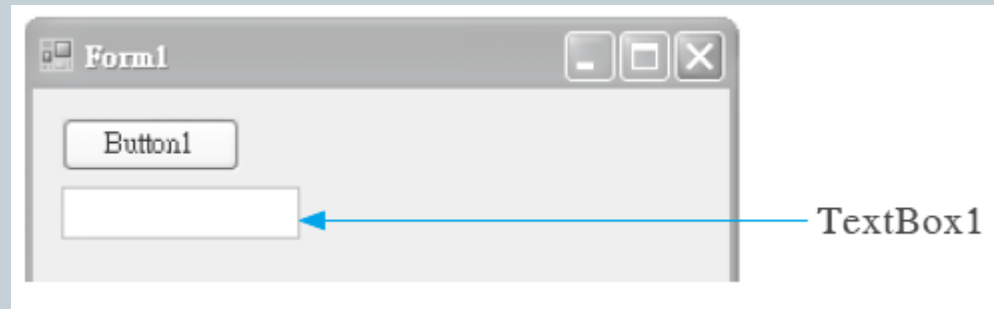
# 選擇排序前言 - 找出陣列中最大值

## ( ch9\找出陣列中最大值\)

- 表單配置

在表單中新增一個

**TextBox**與一個**Button**。



- 屬性設定

物件	屬性	設定值
TextBox1	Multiline	True
TextBox1	Size	200,100
TextBox1	ScrollBars	Both
Button1	Text	找最大值





# 選擇排序前言 - 找出陣列中最大值

## ( ch9\找出陣列中最大值\)

- 程式碼新增與解說

- 點選「找最大值」按鈕，新增程式碼在**Button1\_Click**函式中，如下方程式區塊。

行號	程式碼
3	Dim score() = {60, 90, 44, 98, 50}
4	Dim max As Integer = score(0)
5	TextBox1.Text = TextBox1.Text & "score(0)=" & score(0) & vbNewLine
6	For i = 1 To 4
7	If (max < score(i)) Then
8	max = score(i)
9	End If
10	TextBox1.Text = TextBox1.Text & "score(" & i & ")=" & score(i) & vbNewLine
11	Next
12	TextBox1.Text = TextBox1.Text & "max=" & max

# 選擇排序前言 - 找出陣列中最大值

(  ch9\找出陣列中最大值\)

## ● 解說

- 第3行：宣告陣列**score**為整數，且初始化**score**陣列的第1個元素為**60**、第2個元素為**90**、第3個元素為**44**、第4個元素為**98**、第5個元素為**50**。
- 第4行：宣告變數**max**為整數，且初始化**score**陣列的第1個元素。
- 第5行：於元件**TextBox1**中顯示陣列**score(0)**的值。
- 第6~11行：使用**For**迴圈依序取出第2到5個元素與到目前為止最大值(**max**)比較，若第2到5個元素大於到目前為止最大值(**max**)，則將陣列中元素儲存入到目前為止最大值的變數(**max**)(第7~9行)，印出陣列**score**的每個值到**TextBox1**(第10行)。
- 第12行：顯示找最大值結果，加入**TextBox1**的最後。

## 選擇排序( ch9\選擇排序\)



- 隨機產生一個陣列含五個元素，並將這五個元素由小到大排序。
- 想一想
  - 現在有一簡單的想法，先找出五個元素的最大一個元素，將他與陣列最後一個元素交換，如此我們已經將最大元素放到陣列最後一個元素；
  - 依此類推，將範圍改成第一到第四個元素，照上述方式可以找出四個元素中最大的元素與陣列第四個元素交換；
  - 將範圍改成第一到第三個元素，照上述方式可以找出三個元素中最大的元素與陣列第三個元素交換；
  - 將範圍改成第一到第二個元素，照上述方式可以找出兩個中最大元素與陣列第二個元素交換。
  - 到此已經完成陣列資料的排序，此排序方法為「選擇排序」。

## 選擇排序(ch9\選擇排序\)

### ● 舉例說明

- 假設隨機產生五個陣列元素，如下圖。

60	90	44	82	50
----	----	----	----	----

- 1. 初始化陣列索引`max`指向第一個元素(60)，將第一個元素視為最大值。

<code>max</code> ↓				
60	90	44	82	50

- 2. 比較陣列索引`max`指向元素(60)與陣列第二個元素(90)，若陣列第二個元素大於陣列索引`max`指向元素，陣列索引`max`指向第二個元素。

<code>max</code> ↓				
60	90	44	82	50

## 選擇排序(ch9\選擇排序\)

- 3. 比較陣列索引**max**指向元素(90)與陣列第三個元素(44)，若陣列第三個元素大於陣列索引**max**指向元素，陣列索引**max**指向第三個元素。

60	90	44	82	50
----	----	----	----	----

- 4. 比較陣列索引**max**指向元素(90)與陣列第四個元素(82)，若陣列第四個元素大於陣列索引**max**指向元素，陣列索引**max**指向第四個元素。

60	90	44	82	50
----	----	----	----	----

- 5. 比較陣列索引**max**指向元素(90)與陣列第五個元素(50)，若陣列第五個元素大於陣列索引**max**指向元素，陣列索引**max**指向第五個元素。

60	90	44	82	50
----	----	----	----	----

## 選擇排序(ch9\選擇排序\)

- 6. 將陣列索引|max指向元素(90)與陣列第五個元素(50)交換。

60	50	44	82	90
----	----	----	----	----

- 7. 依此類推，將範圍改成第一到第四個元素，陣列索引|max指向第一個元素，將第一個元素視為最大值。

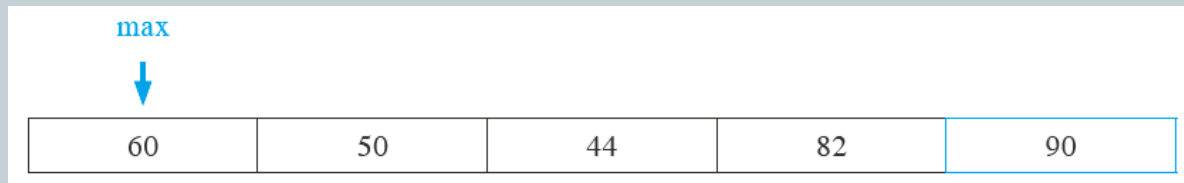
max ↓				
60	50	44	82	90

- 8. 比較陣列索引|max指向元素(60)與陣列第二個元素(50)，若陣列第二個元素大於陣列索引|max指向元素，陣列索引|max指向第二個元素。

max ↓				
60	50	44	82	90

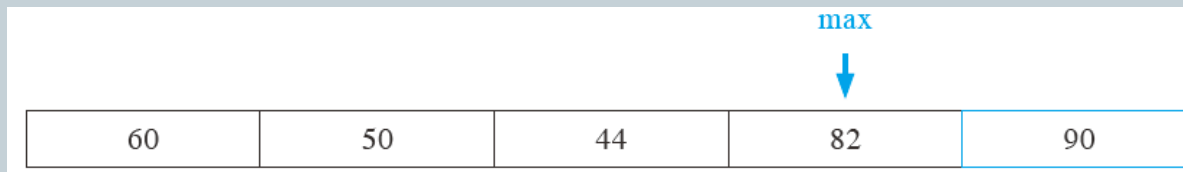
## 選擇排序(ch9\選擇排序\)

- 9. 比較陣列索引`max`指向元素(60)與陣列第三個元素(44)，若陣列第三個元素大於陣列索引`max`指向元素，陣列索引`max`指向第三個元素。



60	50	44	82	90
----	----	----	----	----

- 10. 比較陣列索引`max`指向元素(60)與陣列第四個元素(82)，若陣列第四個元素大於陣列索引`max`指向元素，陣列索引`max`指向第四個元素。



60	50	44	82	90
----	----	----	----	----

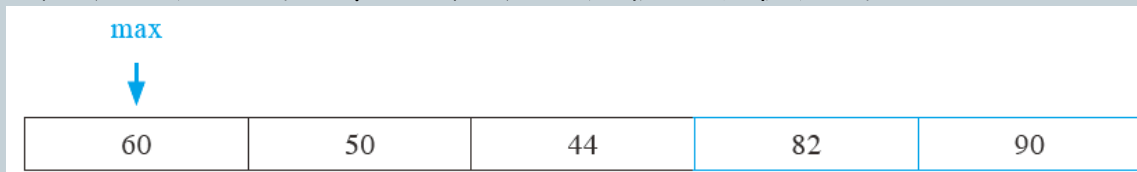
- 11. 將陣列索引`max`指向元素與陣列第四個元素交換。



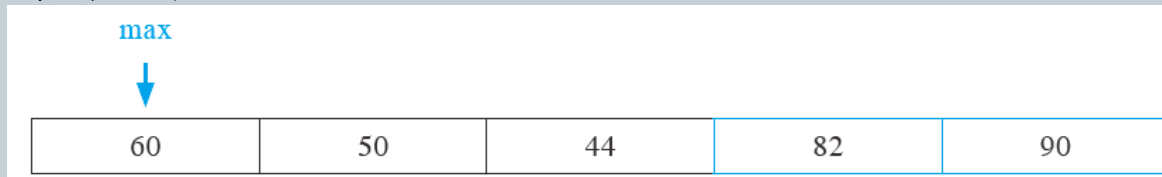
60	50	44	82	90
----	----	----	----	----

## 選擇排序(ch9\選擇排序\)

- 12. 依此類推，將範圍改成第一個到第三個元素，陣列索引`max`指向第一個元素，將第一個元素視為最大值。



- 13. 比較陣列索引`max`指向元素(60)與陣列第二個元素(50)，若陣列第二個元素大於陣列索引`max`指向元素，陣列索引`max`指向第二個元素。



- 14. 比較陣列索引`max`指向元素(60)與陣列第三個元素(44)，若陣列第三個元素大於陣列索引`max`指向元素，陣列索引`max`指向第三個元素。





## 選擇排序(ch9\選擇排序\)

- 15. 將陣列索引`max`指向元素與陣列第三個元素交換。

44	50	60	82	90
----	----	----	----	----

- 16. 依此類推，將範圍改成第一到第二個元素，陣列索引`max`指向第一個元素，將第一個元素視為最大值。

<code>max</code> ↓				
44	50	60	82	90

- 17. 比較陣列索引`max`指向元素(44)與陣列第二個元素(50)，若陣列第二個元素大於陣列索引`max`指向元素，陣列索引`max`指向第二個元素。

<code>max</code> ↓				
44	50	60	82	90

## 選擇排序( ch9\選擇排序\)



- 18. 將陣列索引`max`指向元素與陣列第二個元素交換。

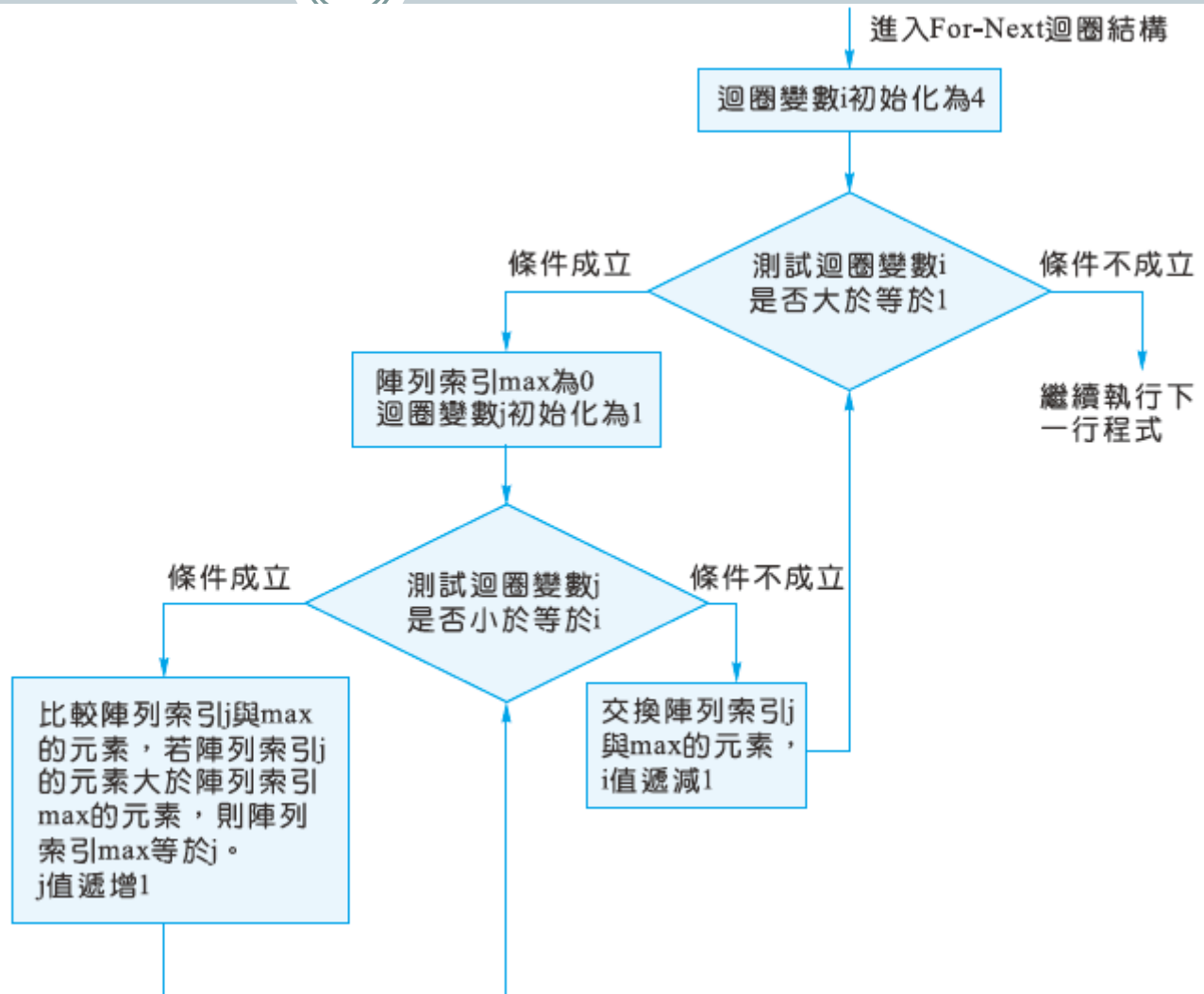
44	50	60	82	90
----	----	----	----	----

- 19. 剩下一個元素程式結束，排序完成。

44	50	60	82	90
----	----	----	----	----

# 選擇排序(ch9\選擇排序\)

- 流程圖表示



# 選擇排序( ch9\選擇排序\)



- 演算法說明

- Step1)此範例需要一個陣列，隨機產生五個數儲存於陣列中，使用巢狀迴圈(For)，外層迴圈變數*i*控制要排序陣列元素的上限範圍，陣列索引**max**指向目前最大元素，初始化為0；
- 內層迴圈變數*j*依序取出陣列每個元素，初始化為1。若內層迴圈變數*j*所指向的陣列元素大於陣列索引**max**所指元素，則陣列索引**max**為內層迴圈變數*j*的值，
- *j*值加1，再次比較陣列索引**max**所指元素與內層迴圈變數*j*所指向的陣列元素，若內層迴圈變數*j*所指向的陣列元素大於陣列索引**max**所指元素，則陣列索引**max**為內層迴圈變數*j*的值，
- 直到外層迴圈變數*i*所限制的上限，可以確保陣列索引**max**指向陣列的最大元素，將陣列索引**max**所指向元素與目前排序範圍的陣列最後一個元素交換，內層迴圈結束。

## 選擇排序( ch9\選擇排序\)

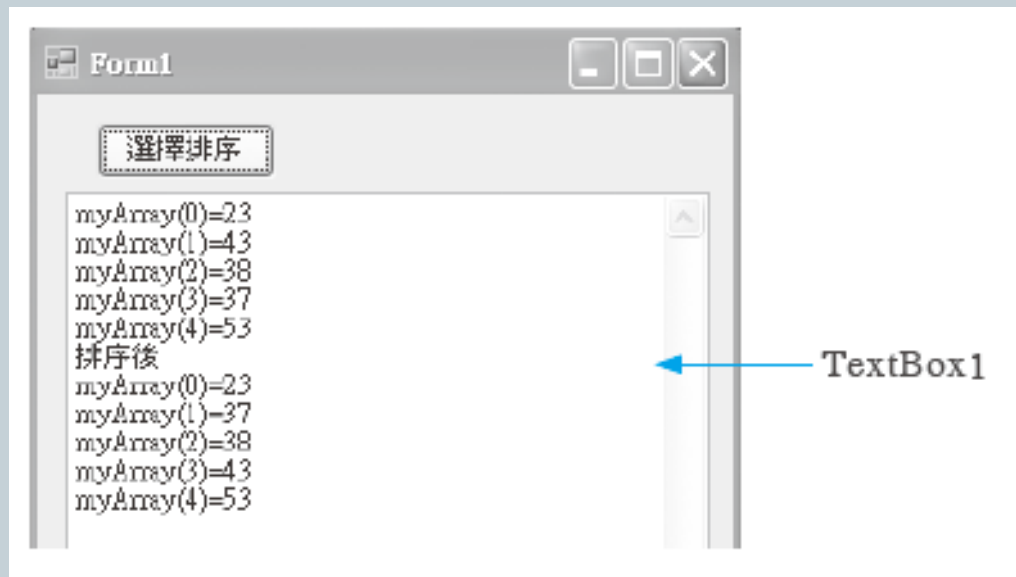


- **Step2)** 外層迴圈變數減1，縮小排序範圍內層迴圈從新開始，陣列索引 $\mathbf{max}$ 指向目前最大元素，初始化為0；內層迴圈變數 $\mathbf{j}$ 依序取出陣列每個元素， $\mathbf{j}$ 初始化為1，若內層迴圈變數 $\mathbf{j}$ 所指向的陣列元素大於陣列索引 $\mathbf{max}$ 所指元素，則陣列索引 $\mathbf{max}$ 為內層迴圈變數 $\mathbf{j}$ 的值，
- $\mathbf{j}$ 值加1，再次比較陣列索引 $\mathbf{max}$ 所指元素與內層迴圈變數 $\mathbf{j}$ 所指向的陣列元素，若內層迴圈變數 $\mathbf{j}$ 所指向的陣列元素大於陣列索引 $\mathbf{max}$ 所指元素，則陣列索引 $\mathbf{max}$ 為內層迴圈變數 $\mathbf{j}$ 的值，直到外層迴圈變數 $\mathbf{i}$ 所限制的上限，可以確保陣列索引 $\mathbf{max}$ 指向陣列的最大元素，將陣列索引 $\mathbf{max}$ 所指向元素與目前排序範圍的陣列最後一個元素交換，可以確保縮小排序範圍中最大元素在最後面。
- **Step3)** 重複上述步驟，縮小排序範圍，將最大元素置於排序範圍的最後面，直到剩下一個元素，就排序完成。

## 選擇排序( ch9\選擇排序\)

- 預覽結果

- 按下「選擇排序」按鈕，結果顯示在**TextBox1**中。



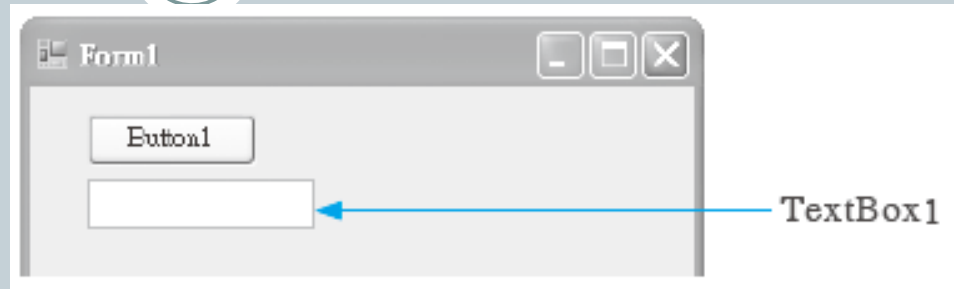
# 選擇排序( ch9\選擇排序\)

- 表單配置

在表單中新增一個  
**TextBox**與一個**Button**。

- 屬性設定

物件	屬性	設定值
Button1	Text	選擇排序
TextBox1	ScrollBars	Both
TextBox1	Multiline	True
TextBox1	Size	270,215



## 選擇排序( ch9\選擇排序\)

### ○ 程式碼新增與解說

- ★ 點選「選擇排序」按鈕，新增程式碼在 **Button1\_Click** 函式中，如下方程式區塊。

行號	程式碼
3	Dim myArray(5) As Integer
4	Dim temp As Integer
5	Dim max As Integer
6	Randomize()
7	For i As Integer = 0 To 4
8	myArray(i) = CInt(Int(Rnd() * 100))
9	TextBox1.Text = TextBox1.Text & "myArray(" & i & ")=" & myArray(i) & vbNewLine
10	Next
11	For i As Integer = 4 To 1 Step -1
12	max = 0
13	For j As Integer = 1 To i
14	If myArray(j) > myArray(max) Then
15	max = j
16	End If
17	Next
18	temp = myArray(i)
19	myArray(i) = myArray(max)
20	myArray(max) = temp
21	Next
22	TextBox1.Text = TextBox1.Text & "排序後a" & vbNewLine
23	For i As Integer = 0 To 4
24	TextBox1.Text = TextBox1.Text & "myArray(" & i & ")=" & myArray(i) & vbNewLine



# 選擇排序( ch9\選擇排序\)



- 解說

- 第3行：宣告五個元素的整數陣列**myArray**。
- 第4行：宣告**temp**為整數變數。
- 第5行：宣告**max**為整數變數，為陣列索引指向陣列目前最大元素。
- 第6行：初始化隨機函式。
- 第7到10行：使用**For**迴圈隨機產生陣列**myArray**元素的值，其值為介於0到99的整數。顯示陣列**myArray**元素值於**TextBox1**。

## 選擇排序( ch9\選擇排序\)



- 解說

- 第11到21行：選擇排序演算法，外層迴圈變數*i*，控制內層迴圈變數*j*的上限，迴圈變數*i*由4到1，每次遞減1。
- 第12行：初始化陣列索引|max為0。
- 第13到17行：內層迴圈*j*由1到*i*，每次遞增1。
- 第14到16行：比較陣列索引*j*與陣列索引|max，若陣列索引*j*元素值大於陣列索引|max元素值，則陣列索引|max令為*j*，內層迴圈結束後，|max指向陣列myArray第0個到第*i*個中最大的元素。
- 第18到20行：陣列索引|max之元素與陣列索引|i的元素交換。
- 第22行：顯示「排序後」於TextBox1。
- 第23到25行：顯示排序後陣列所有元素。