

ជំពូកទី១ :

Arrays & Collections

នៅក្នុងមេរៀននេះលោកអ្នកនឹងសិក្សាពី :

👉 វិមាត្ររបស់ Arrays

👉 ប្រភេទនៃ Arrays

👉 របៀបប្រកាស Arrays

👉 ការប្រើប្រាស់ Collections

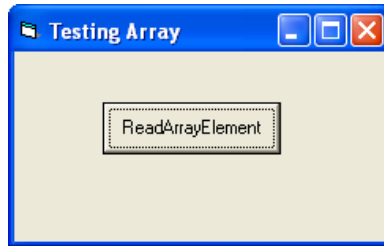
👉 ការប្រើ For Each . . . Next ជាមួយនឹង Collection's Object

កាលពីនៅក្នុងមេរៀនមុន លោកអ្នកបានសិក្សារួចមកហើយពីរបៀបប្រកាសអថេរ ដែលរាល់អថេរទាំងនោះគឺ សុទ្ធតែអាចរក្សាតំលៃ បានតែមួយគត់។ ចំណែកនៅក្នុងមេរៀននេះវិញលោកអ្នកនឹងរៀនពីការបង្កើត និង ប្រើអថេរមួយប្រភេទដែលអនុញ្ញាតអោយ លោកអ្នកបញ្ចូលតំលៃជាទំរង់សំនុំតំលៃទៅអោយអថេរទាំងនោះបាន។ ដើម្បីអោយអថេរមួយអាចទទួលយកតំលៃជាសំនុំតំលៃបានលោកអ្នកត្រូវប្រកាសអថេរ នោះជា Array ។

ឧទាហរណ៍ :

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី៦.1 ។

រូបភាពទី៦.១



☞ Source Code:

```
Option Explicit
Dim A(3) As String
```

```
.....
Private Sub Form_Load( )
    A(0) = "VB"
    A(1) = "The best"
    A(2) = "Programming for"
    A(3) = "Window Application"
End Sub
```

```
.....
Private Sub CmdReadElement_Click()
    MsgBox A(1) & vbCrLf & A(3)
End Sub
.....
```

✍ **វិភាគ :**

- លោកអ្នកបានឃើញហើយថានៅក្នុងឧទាហរណ៍ខាងលើ យើងបានធ្វើការប្រកាសអថេរមួយគឺ A(3) ដែលមានប្រភេទទិន្នន័យជា Array នៃ String ហើយមាន Index ចំនួនបួនគឺពីលេខ 0 ដល់ លេខ 3 សំរាប់រក្សាទុកទិន្នន័យ។ ហើយចំពោះអថេរ A នេះគឺវាមានលក្ខណៈ ដូចជាអថេរធម្មតាដែរ ប៉ុន្តែ វាខុសពីអថេរធម្មតាត្រង់ថា វាអាចរក្សាទុកតំលៃទិន្នន័យបានចំនួនបួនទីតាំងទៅតាម Index របស់វាដូចដែលលោកអ្នកបានឃើញនៅក្នុងឧទាហរណ៍ ខាងលើ។

- លោកអ្នកអាចធ្វើការបញ្ចូលទិន្នន័យ ឬ អានទិន្នន័យចេញពីអថេរ A នេះបាន តាមរយៈ Index របស់វា ដូចនៅក្នុងឧទាហរណ៍ខាងលើ។

I. វិមាត្ររបស់ Array (Dimensional Array)

ឥឡូវនេះយើងងាកមកនិយាយពីវិមាត្ររបស់ Array វិញម្តង។ ចំពោះវិមាត្ររបស់ Array ត្រូវបានចែកចេញជាពីរ គឺ Array មួយវិមាត្រ (One Dimensional Array) និង Array ច្រើនវិមាត្រ (Multiple Dimensional Array) ។

I.1. Array មួយវិមាត្រ (One Dimensional Array)

- នៅក្នុង Array មួយវិមាត្រត្រូវបានចែកចេញជាពីរប្រភេទទៀតគឺ :
- ◆ Fixed Size Array
 - ◆ Dynamic Array

I.1.2. Fixed Size Array

គឺជាប្រភេទ Array ដែលត្រូវបានកំណត់ចំនួនទីតាំងនៅក្នុង Memory ច្បាស់លាស់នៅក្នុងពេលប្រកាសអថេរ។

I.1.2.1. របៀបប្រកាស Fixed Size Array

Syntax:

```
Private / Public/ Dim ArrayName (UBound ) As DataType
```

សមាសភាពនៃវត្ថុ Syntax	អធិប្បាយ
ArrayName	ជាអថេរដែលមានប្រភេទជា Array ទៅតាមប្រភេទទិន្នន័យនៃ DataType
UBound	ជាទីតាំងដំបូងនៃ Index របស់ Array ដែលលោកអ្នកត្រូវកំណត់ទុកជាមុនដើម្បី អោយ ArrayName អាចរក្សាទុកទិន្នន័យ បាននៅក្នុង Memory ។

ឧទាហរណ៍ :

☞ Source Code:

```
Option Explicit
Dim A(5) As String
```

```
.....
Private Sub Form_Load()
    A(1) = "BBU"
    A(3) = "Build Bright University"
End Sub
.....
```

✍ **វិភាគ :**

បន្ទាប់ពីលោកអ្នកបានប្រកាស Array ដែលមានឈ្មោះ A និង UBound=5 រួចមក នៅពេលដែល Code Dim A(5) As String ត្រូវបានដំណើរការ ពេលនោះនឹងមាន ការចាប់ផ្តើមកំណត់ទីតាំងដែលមាននៅក្នុង Memory ទាំងអស់ចំនួន 6 ទីតាំង គឺពីទីតាំង ទី 0 ដល់ ទីតាំងទី៥ ដើម្បីរក្សាទុកទិន្នន័យប្រភេទជា String ដែលមាន ទំរង់ ដូចខាងក្រោម :

0	1	2	3	4	5
	BBU		Build Bright University		

☞ នៅក្នុងភាសា Visual Basic Index របស់ Array ត្រូវបានចែកជាពីរគឺ

◆ **LBound Index :**

ជា Index ដែលតូចជាងគេបង្អស់របស់ Array ។ ដើម្បី ទាញយក Index តូចបង្អស់របស់ Array មួយលោកអ្នកត្រូវប្រើអនុគមន៍ មួយឈ្មោះ LBound ។

Syntax:

LBound(ArrayName)

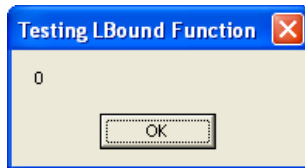
ឧទាហរណ៍ :

☞ Source Code:

Option Explicit

```
Private Sub Form_Load( )
    Dim A(5) As String
    Dim Indx As Integer
    Indx = LBound(A)
    MsgBox Indx, , "Testing LBound Function"
End Sub
```

លទ្ធផល :



◆ UBound Index :

ជា Index ដែលធំជាងគេបង្អស់របស់ Array ។ ដូច្នេះដើម្បី ទាញយក តំលៃ Index របស់ Array ដែលមានតំលៃធំជាងគេលោកអ្នក ត្រូវប្រើអនុគមន៍មួយឈ្មោះ UBound ។

Syntax:

UBound(ArrayName)

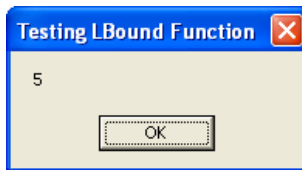
ឧទាហរណ៍ :

☞ Source Code:

Option Explicit

```
Private Sub Form_Load()  
    Dim A(5) As String  
    Dim Indx As Integer  
    Indx = UBound(A)  
    MsgBox Indx, , "Testing LBound Function"  
End Sub
```

លទ្ធផល:



I.1.2.2. របៀបប្រកាស Fixed Size Array ដោយកំណត់ LBound និង UBound

Syntax:

```
Dim ArrayName(LBound To UBound) As DataType
```

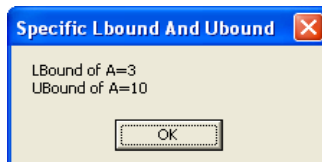
ឧទាហរណ៍ :

☞ Source Code:

Option Explicit

```
Private Sub Form_Load()  
    Dim A(3 To 10) As String  
    MsgBox "LBound of A=" & LBound(A) & vbCrLf & "UBound of A=" &  
        UBound(A), , "Specific Lbound And Ubound "  
End Sub
```

លទ្ធផល:



I.1.2.3. ការប្រើ Option Base 1

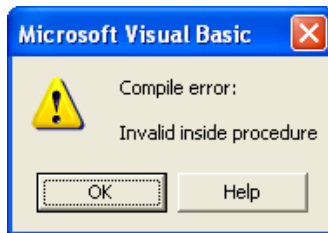
ចំពោះការប្រើ Option Base 1 KeyWord នេះគឺដើម្បីកំណត់ Index ដំបូង បង្អស់របស់ Array អោយមានតំលៃស្មើអោយ ១។ ហើយរបៀបប្រើវា គឺតំរូវអោយលោក អ្នកយកពាក្យថា Option Base 1 នេះទៅប្រកាសនៅផ្នែក General Declaration បើមិន ដូច្នោះទេនឹងមាន Error កើតឡើងដូចរូបភាពទី៦.២។

ឧទាហរណ៍ទី១ :

☞ Source Code:
Option Explicit

```
Private Sub Form_Load()  
Option Base 1  
Dim A(5) As String  
MsgBox "LBound of A=" & LBound(A)  
MsgBox "UBound of A=" & UBound(A)  
End Sub
```

រូបភាពទី៦.២

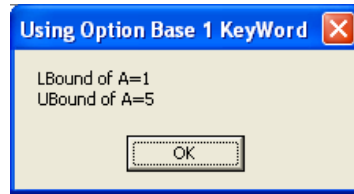


ឧទាហរណ៍ទី២ :

☞ Source Code:
Option Explicit
Option Base 1

```
Private Sub Form_Load()  
Dim A(5) As String  
MsgBox "LBound of A=" & LBound(A) & vbCrLf & "UBound of A=" &  
UBound(A), , "Using Option Base 1 KeyWord"  
End Sub
```

លទ្ធផល :



ចំណាំ:

ជាទូទៅការប្រើ Option Base 1 KeyWord មិនមានឥទ្ធិពលទៅលើការប្រកាស Array ដោយធ្វើការកំណត់ LBound និង UBound ជាមុននោះទេ។

ឧទាហរណ៍ទី៣ :

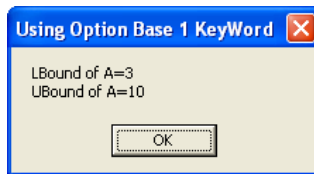
☞ Source Code:

Option Explicit
Option Base 1

```

Private Sub Form_Load()
    Dim A(3 To 10) As String
    MsgBox "LBound of A=" & LBound(A) & vbCrLf & "UBound of A=" &
        UBound(A), , "Using Option Base 1 KeyWord"
End Sub
  
```

លទ្ធផល :



1.1.2.4. ការបញ្ចូលតំលៃទៅអោយធាតុរួម Fixed Size Array

Syntax:

ArrayName(Index) = Value

ឧទាហរណ៍ :

☞ Source Code:

Option Explicit

Option Base 1
Dim A(4) As String

```
Private Sub Form_Load()  
    A(1) = "You are still young."  
    A(2) = "You can drive."  
    A(3) = "You can drink"  
    A(4) = "Time has come to retire"  
End Sub
```

I.1.2.5. ការទាញយកតំលៃពីធាតុរបស់ Fixed Size Array

Syntax:

VariableName= ArrayName(Index)

ឧទាហរណ៍ទី១ :

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី៦.១។

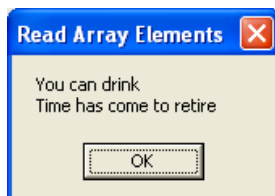
☞ Source Code:

```
Option Explicit  
Option Base 1  
Dim A(4) As String
```

```
Private Sub Form_Load( )  
    A(1) = "You are still young."  
    A(2) = "You can drive."  
    A(3) = "You can drink"  
    A(4) = "Time has come to retire"  
End Sub
```

```
Private Sub CmdReadElement_Click()  
    MsgBox A(3) & vbCrLf & A(4), , "Read Array Elements"  
End Sub
```

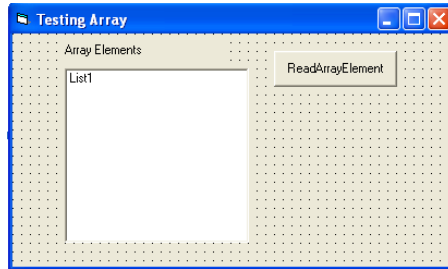
លទ្ធផល :



ឧទាហរណ៍ទី២ :

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.៣។

រូបភាពទី៦.៣



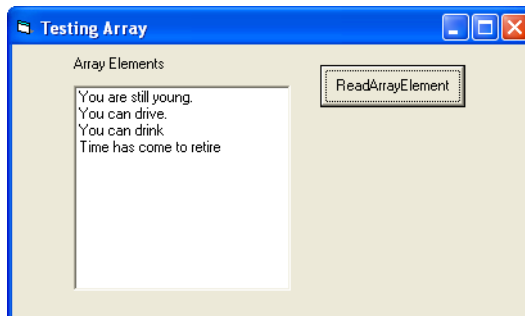
☞ Source Code:

```
Option Explicit  
Option Base 1  
Dim A(4) As String
```

```
Private Sub Form_Load( )  
    A(1) = "You are still young."  
    A(2) = "You can drive."  
    A(3) = "You can drink"  
    A(4) = "Time has come to retire"  
End Sub
```

```
Private Sub CmdReadElement_Click()  
    Dim i%  
    For i = LBound(A) To UBound(A)  
        Me.List1.AddItem A(i)  
    Next  
End Sub
```

លទ្ធផល :



I.1.3. Dynamic Array

ជាប្រភេទ Array ដែលមិនមានការកំណត់ចំនួន UBound នៅពេលប្រកាស ហើយ តំលៃ UBound របស់វាអាចប្រែប្រួលបាននៅពេល RunTime ។

I.1.3.1. របៀបប្រកាស

Syntax:

```
Dim ArrayName( ) As
```

ឧទាហរណ៍ :


```
Dim A( ) As String
```

ចំពោះការប្រើ Dynamic Array ពេលប្រកាសគេមិនប្រើ Ubound នោះទេ ពីព្រោះ Ubound ត្រូវបានប្រើនៅពេលប្រកាស Array ឡើងវិញ។ ដើម្បីប្រកាស Array ឡើងវិញ លោកអ្នកអាចប្រើ KeyWord មួយគឺ Redim ឬ Redim Preserve ។

I.1.3.2. ការប្រើ Redim KeyWord


Syntax:

```
Redim ArrayName(UBound)
```

 **ចំណាំ :**

ចំពោះការប្រើ Redim KeyWord គឺវាមិនអាចរក្សាតំលៃនៃធាតុរបស់ Array បានឡើយព្រោះនៅពេលដំណើរការជួប Redim KeyWord ពេលនោះបណ្តាលអោយ Index របស់ Array ត្រលប់មកកាន់ទីតាំងដើម (LBound) វិញដែលជាមូលហេតុធ្វើអោយតំលៃនៃធាតុរបស់វាបាត់បង់។

ឧទាហរណ៍ទី១ :

 Source Code:

```
Option Explicit
Dim A( ) As String
```

```
.....
Private Sub Form_Load()
    ReDim A(2)
```

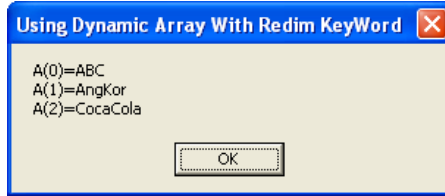
```

A(0) = "ABC"
A(1) = "AngKor"
A(2) = "CocaCola"
MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf & "A(2)="
& A(2), , "Using Dynamic Array With Redim KeyWord"

```

End Sub

លទ្ធផល:



បន្ទាប់ពីលោកអ្នកបានប្រកាស Dynamic Array ដែលមានឈ្មោះ A រួចហើយ ដោយមានប្រើនូវ Redim A(2) KeyWord នៅក្នុង Event Form_Load () នៅពេលដែលដំណើរការជួប Redim KeyWord នោះ Compiler នឹងបញ្ជាអោយបង្កើត Location Memory ចំនួន ៣ ទីតាំងសំរាប់ទុកតំលៃនៃធាតុរបស់ Array A ដែលមាន ទំរង់ដូចខាងក្រោម:

0	1	2
ABC	AngKor	CocaCola

ឧទាហរណ៍ទី២ :

☞ Source Code:

```

Option Explicit
Dim A() As String

```

```

Private Sub Form_Load()
    ReDim A(2)
    A(0) = "ABC"
    A(1) = "AngKor"
    A(2) = "CocaCola"
    MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf &
"A(2)=" & A(2), , "Using Dynamic Array With Redim
KeyWord"
    ReDim A(4)

```

```

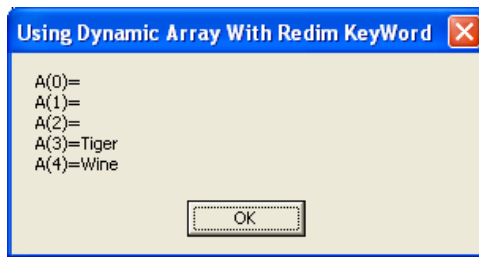
A(3) = "Tiger"
A(4) = "Wine"
MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf &
"A(2)=" & A(2) & vbCrLf & "A(3)=" & A(3) & vbCrLf &
"A(4)=" & A(4), , "Using Dynamic Array With Redim
Keyword"

```

End Sub

លទ្ធផលបង្ហាញពីស្រ្តី Redim KeyWord លើកទី១ គឺដូចរូបភាពទី៦.៤។

លទ្ធផលបង្ហាញពីស្រ្តី Redim KeyWord លើកទី២ :



ចំពោះឧទាហរណ៍ទី២ នេះចាប់ពីបន្ទាត់ទី១ ដល់ បន្ទាត់ដែលមាន Code MsgBox MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf & "A(2)=" & A(2), , "Using Dynamic Array With Redim Keyword" គឺមានលក្ខណៈដូចគ្នាទៅនឹង ឧទាហរណ៍ទី១ ប៉ុន្តែចាប់ពី បន្ទាត់នេះបន្តទៀត លោកអ្នក បានឃើញការប្រើ Redim A(4) ម្តងទៀតដើម្បីប្រកាស Array សារជាថ្មីឡើងវិញ ដែលធ្វើអោយ តំលៃចាស់របស់វា បាត់បង់។ នៅពេលដែល ប្រកាស Array សារឡើងវិញ វាមានទំរង់ដូចខាងក្រោម:

Array ចាស់ : Redim A(2)

0	1	2
ABC	AngKor	CocaCola

Array ថ្មី : Redim A(4)

0	1	2	3	4
			Tiger	Wine

I.1.3.3. ការប្រើ Redim Preserve KeyWord

Syntax:

Redim Preserve ArrayName(UBound)

ចំពោះការប្រើ Redim Preserve វាអាចរក្សាតំលៃនៃធាតុរបស់ Array បាន ព្រោះនៅពេលដែលជួប Keyword Redim Preserve ពេលនោះ Index របស់វាតំបន់ កាន់ទីតាំងចុងក្រោយហើយទើបបន្ថែម Location បន្តទៀត។

ឧទាហរណ៍ទី១ :

☞ Source Code:

Option Explicit
Dim A() As String

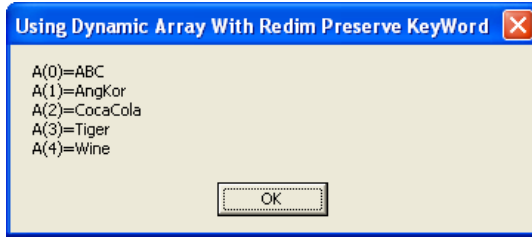
```

Private Sub Form_Load()
    ReDim A(2)
    A(0) = "ABC"
    A(1) = "AngKor"
    A(2) = "CocaCola"
    MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf &
        "A(2)=" & A(2), , "Using Dynamic Array With Redim
        KeyWord"
    ReDim Preserve A(4)
    A(3) = "Tiger"
    A(4) = "Wine"
    MsgBox "A(0)=" & A(0) & vbCrLf & "A(1)=" & A(1) & vbCrLf &
        "A(2)=" & A(2) & vbCrLf & "A(3)=" & A(3) & vbCrLf &
        "A(4)=" & A(4), , "Using Dynamic Array With Redim
        Preserve KeyWord"
End Sub

```

លទ្ធផលបង្ហាញពីប្រើ Redim KeyWord លើកទី១ គឺដូចរូបភាពទី៦.៤។

លទ្ធផលបង្ហាញពីប្រើ Redim Preserve KeyWord



ទំរង់របស់ Array A ដូចខាងក្រោម:

Array ទាស់ : Redim A(2)

	0	1	2
ABC	AngKor	CocaCola	

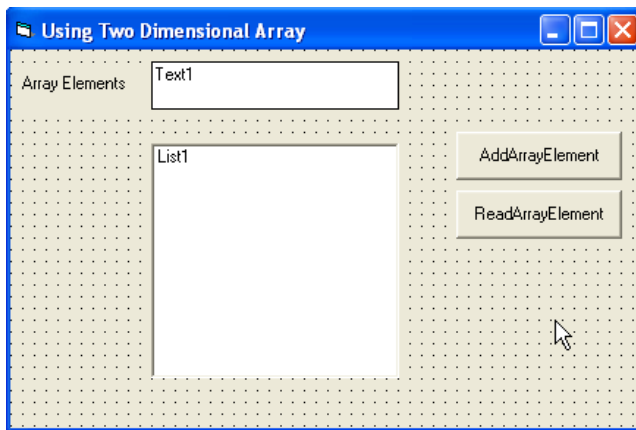
Array ថ្មី : Redim Preserve A(4)

	0	1	2	3	4
ABC	AngKor	CocalCola	Tiger	Wine	

ឧទាហរណ៍ទី២ :

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី៦.៥។

រូបភាពទី៦.៥



គោរម៍ណង :

- ពេលដែលលោកអ្នកចុចលើ AddArrayElement Button សូមអោយយក

ទិន្នន័យដែលមាននៅក្នុង Text1 ទៅចូលក្នុង Array ។

- ពេលដែលលោកអ្នកចុចលើ ReadArrayElement សូមទាញទិន្នន័យក្នុង ធាតុរបស់ Array

ទាំងអស់មកដាក់ក្នុង List1 ។

☞ Source Code:

```
Option Explicit
Dim ArrValue() As String
Dim Indx As Integer
```

```
.....
Private Sub CmdAddArrayElement_Click()
    ReDim Preserve ArrValue(Indx)
    ArrValue(Indx) = Me.Text1
    Indx = Indx + 1
End Sub
```

```
.....
Private Sub CmdReadElement_Click()
    ReadArrValue ArrValue, Me.List1
End Sub
```

```
.....
Sub ReadArrValue(Arr() As String, Lstbox As ListBox)
    Dim i%
    Lstbox.Clear
    For i = LBound(Arr) To UBound(Arr)
        Lstbox.AddItem Arr(i)
    Next
End Sub
.....
```

I.2. Array ច្រើនវិមាត្រ (Multiple Dimensional Array)

ចំពោះ Array ច្រើនវិមាត្រក៏មានពីរប្រភេទដូចគ្នាគឺ Fixed Size Array និង Dynamic Array ។

I.2.1. Array ពីរវិមាត្រ (Two Dimensional Array)

I.2.1.1. Fixed Size Array

I.2.1.1.1. ប្រើប្រាស់

Syntax:

```
Dim ArrayName(URow,UColumn) As DataType
```

- URow ជាចំនួន UBound Rows នៃ Index ធំជាងគេនៅក្នុងជួរដេក។
- UColumn ជាចំនួន UBound Columns នៃ Index ធំជាងគេនៅក្នុង ជួរឈរ។

ឧទាហរណ៍:

Dim A(2, 3) As String នោះ Array មានទំហំដូចខាងក្រោម:

	0	1	2	3
0				
1				
2				

I.2.1.1.2. ប្រើប្រាស់ដោយកំណត់ LBound និង UBound

Syntax:

```
Dim ArrayName(LRow To URow,LCol To UCol) As  
DataType
```

ឧទាហរណ៍:

Dim A(1 To 2, 1 To 3) As String នោះ Array មានទំហំដូចខាងក្រោម:

	1	2	3
1			
2			

I.2.1.1.2. ការប្រើ LBound និង UBound Funtion ជាមួយនឹង Array ពីរវិមាត្រ

Syntax សំរាប់ទាញយក Index នៃជួរដេកតូចបំផុតរបស់ Array ពីរវិមាត្រ :

```
LBound(ArrayName,1)
```

Syntax សំរាប់ទាញយក Index នៃជួរដេកធំបំផុតរបស់ Array ពីរវិមាត្រ :

```
UBound(ArrayName,
```

Syntax សំរាប់ទាញយក Index នៃជួរឈរតូចបំផុតរបស់ Array ពីរិមាត្រ :

`LBound(ArrayName, 2)`

Syntax សំរាប់ទាញយក Index នៃជួរឈរធំបំផុតរបស់ Array ពីរិមាត្រ :

`UBound(ArrayName,2)`

ចំណាំ

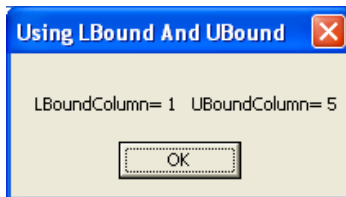
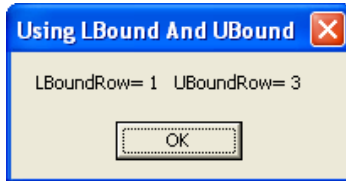
- លេខ 1 នៃ Argument ទី២ តំណាងអោយជួរដេក (Row)
- លេខ 2 នៃ Argument ទី២ តំណាងអោយជួរឈរ (Column)

ឧទាហរណ៍ :

☞ Source Code:
 Option Explicit
 Option Base 1

```
Private Sub Form_Load()
  Dim A(3, 5) As Integer
  MsgBox "LBoundRow= " & LBound(A, 1) & Space(3) & "UBoundRow= " &
    UBound(A, 1), , "Using LBound And UBound"
  MsgBox vbCrLf & "LBoundColumn= " & LBound(A, 2) & Space(3) &
    "UBoundColumn= " & UBound(A, 2), , "Using LBound
  And UBound"
End Sub
```

លទ្ធផល:



1.2.1.1.3. របៀបបញ្ចូលតំលៃទៅអោយធាតុរបស់ Array ពីវិមាត្រ

Syntax:

```
ArrayName(RowIndex,ColIndex)=Valu
```

ឧទាហរណ៍ :

☞ Source Code:

Option Explicit

Option Base 1

Private Sub Form_Load()

Dim A(3, 5) As String

A(1, 1) = "Tiger"

A(1, 3) = "ABC"

A(2, 2) = "Wine"

A(2, 5) = "Love"

A(3, 1) = "Angkor"

End Sub

ទំរង់ដែល Array ពីវិមាត្រនៃឧទាហរណ៍នេះរក្សាទិន្នន័យគឺដូចតារាងខាងក្រោម :

	1	2	3	4	5
1	Tiger		ABC		
2		Wine			Love
3	Angkor				

2.1.1.4. របៀបទាញយកទិន្នន័យពីធាតុរបស់ Array ពីវិមាត្រ

Syntax:

```
VariableName=ArrayName(RowIndex,ColIndex)
```

ឧទាហរណ៍ :

ឧបមថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី ៦.៦។

☞ Source Code:

Option Explicit

Option Base 1

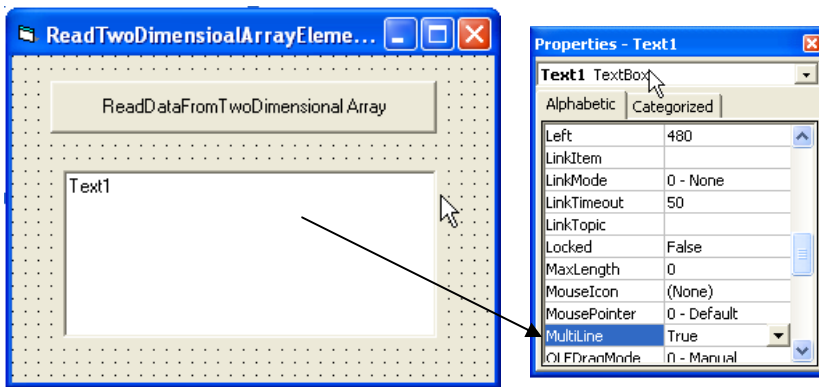
Dim A(3, 5) As String

```

Private Sub Form_Load()
    A(1, 1) = "Tiger"
    A(1, 3) = "ABC"
    A(2, 2) = "Wine"
    A(2, 5) = "Love"
    A(3, 1) = "Angkor"
End Sub

```

រូបភាពទី ៦. ៦



```

Private Sub CmdReaddata_Click()
    Me.Text1 = ReadDataFromTwoDimensionalArray(A)
End Sub

```

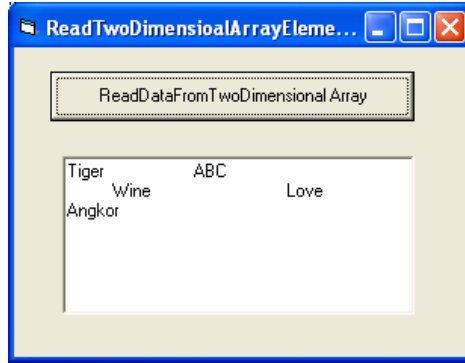
```

Function ReadDataFromTwoDimensionalArray(ArrValue() As String) As String
    Dim i%, j%
    Dim st$
    For i = LBound(ArrValue, 1) To UBound(ArrValue, 1)
        For j = LBound(ArrValue, 2) To UBound(ArrValue, 2)
            If ArrValue(i, j) <> "" Then
                st = st & ArrValue(i, j) & Space(10)
            Else
                st = st & Space(10)
            End If
        Next
        st = st & vbCrLf
    Next
End Function

```

```
ReadDataFromTwoDimensionalArray = st
End Function
```

លទ្ធផល :



I.2.1.2. Dynamic Array

ចំពោះ Dynamic នៃ Two Dimensional Array វិញ របៀបប្រកាសរបស់វាក៏ ដូចគ្នាទៅនឹង One Dimensional Array ដែរ។ ហើយនៅពេលប្រកាស Index Array ឡើងវិញក៏នៅតែប្រើ Redim ឬ Redim Preserve Keyword ដដែល គ្រាន់តែ Index ដែលអាច ប្រែប្រួលបានគឺមានតែ Index របស់ Columns តែប៉ុណ្ណោះ។

ឧទាហរណ៍ទី១ :

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.៦ ដដែល។

☞ Source Code:

```
Option Explicit
Option Base 1
Dim A() As String
```

```
Private Sub Form_Load()
    ReDim A(3, 5)
    A(1, 1) = "Tiger"
    A(1, 3) = "ABC"
    A(2, 2) = "Wine"
    A(2, 5) = "Love"
    A(3, 1) = "Angkor"
End Sub
```

```
Private Sub CmdReadData_Click()
    ReDim Preserve A(3, 7)
```

```

A(3, 6) = "CocaCola"
A(3, 7) = "Pepsi"
Me.Text1 = ReadDataFromTwoDimensionalArray(A)
End Sub

```

◆ ទំរង់ Array A បន្ទាប់ពីប្រើ Redim KeyWord និង បញ្ចូលតំលៃទៅដោយវា បន្ទាប់ពី Form_Load Event ដំណើរការ។

	1	2	3	4	5
1	Tiger		ABC		
2		Wine			Love
3	Angkor				

◆ ទំរង់ Array A បន្ទាប់ពីប្រើ Redim Preserve KeyWord និង បញ្ចូលតំលៃទៅ ដោយ Array បន្ទាប់ពី CmdReadData_Click Event ដំណើរការ។

	1	2	3	4	5	6	7
1	Tiger		ABC				
2		Wine			Love		
3	Angkor					CocaCola	Pepsi

ឧទាហរណ៍ទី២ :

ឧបមាថាលោកអ្នកធ្វើការកែប្រែ Source Code នៅក្នុងឧទាហរណ៍ទី១ ដោយប្រើ Redim Preserve KeyWord ដើម្បីបង្កើន Row បន្ថែម ដូច Source Code ខាងក្រោម ពេលនោះនឹងមាន Error កើតឡើងដូច រូបភាពទី៦.៧។

```

☞ Source Code:
Option Explicit
Option Base 1
Dim A() As String

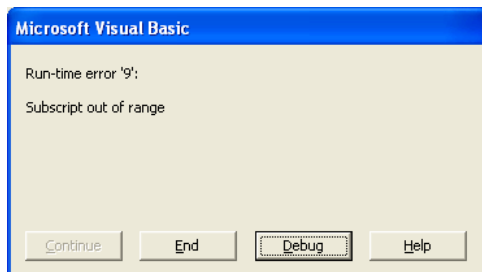
```

```

Private Sub CmdReadData_Click()
ReDim Preserve A(7, 7)
A(3, 6) = "CocaCola"
A(3, 7) = "Pepsi"
Me.Text1 = ReadDataFromTwoDimensionalArray(A)
End Sub

```

រូបភាពទី៦.៧



II. ការប្រើ Collections

នៅក្នុងភាសា Visual Basic ផ្តល់នូវ Object មួយប្រភេទដែលអាចអោយលោកអ្នកបញ្ចូលជាសំនុំទិន្នន័យទៅអោយអថេរដូចជា Array ដែរគឺ Collection ។ លោកអ្នកអាចប្រើនូវ Collection Object ជំនួសអោយ Array បានព្រោះថា វាមានលក្ខណៈងាយស្រួលជាង Array ត្រង់ថាវាមាន Properties និង Methods មកជាមួយដែលអនុញ្ញាតអោយលោកអ្នកធ្វើការជាមួយវាបានយ៉ាងងាយស្រួល។

II.1. ការប្រកាស អថេរវាប្រភេទ Collection

Syntax:

```
Dim ObjVariableName As New
```

ឧទាហរណ៍ :


```
Dim Obj As New Collection
```

II.2. ការបញ្ចូលទិន្នន័យទៅអោយអថេរវាប្រភេទ Collections

Syntax:


```
Object.Add item, [key], [before], [after]
```

Arguments	អធិប្បាយ
Item	ជាទិន្នន័យដែលលោកអ្នកត្រូវបញ្ចូល។
[key]	ជា key តំណាងអោយទិន្នន័យ (item) និមួយៗ ដែលលោកអ្នក ត្រូវបញ្ចូល។
[befor]	ជា Index ឬ key ដែលលោកអ្នកត្រូវ កំណត់ទីតាំងពីខាងមុខនៃទីតាំងនៃទិន្នន័យណា មួយដែលមានស្រាប់សំរាប់ បញ្ចូលទិន្នន័យថ្មី ទៅក្នុង Collection Object ។
[after]	ជា Index ឬ key ដែលលោកអ្នកត្រូវ កំណត់ទីតាំងពីខាងក្រោយនៃទីតាំងនៃទិន្នន័យណា មួយដែលមានស្រាប់សំរាប់ បញ្ចូលទិន្នន័យថ្មី ទៅក្នុង Collection Object ។

 **ចំណាំ :**

នៅក្នុងចំណោម [befor] និង [after] នៃ Arguments ទាំងពីរគឺលោកអ្នកអាចប្រើ បានតែមួយគត់។

ឧទាហរណ៍ទី១ :

 Source Code:

```
Option Explicit
Dim Obj As New Collection
```

```
.....
Private Sub Form_Load( )
    Obj.Add "Srey Phean"
    Obj.Add "Koub Sopheak"
    Obj.Add "To SokNang"
    Obj.Add "Kim SaoRin"
End Sub
.....
```

អំរង់នៃ Obj Collection Object ដែលរក្សាទិន្នន័យគឺ :

	Obj Object
1	Srey Phean
2	Koub Sopheak
3	To SokNang
4	Kim SaoRin

ឧទាហរណ៍ទី២ :

☞ Source Code:

```
Option Explicit
Dim Obj As New Collection
```

```
Private Sub Form_Load()
    Obj.Add "Srey Phean"
    Obj.Add "Koub Sopheak"
    Obj.Add "To SokNang"
    Obj.Add "Kim SaoRin"
    Obj.Add "Khon Sothea", , 3
End Sub
```

និរទ័ន Obj Collection Object ដែលរក្សាទិន្នន័យគឺ :

	Obj Object
1	Srey Phean
2	Koub Sopheak
3	<i>Khon Sothea</i>
4	To SokNang
5	Kim SaoRin

ឧទាហរណ៍ទី៣ :

☞ Source Code:

```
Option Explicit
Dim Obj As New Collection
```

```
Private Sub Form_Load()
    Obj.Add "Srey Phean"
    Obj.Add "Koub Sopheak"
    Obj.Add "To SokNang"
    Obj.Add "Kim SaoRin"
    Obj.Add "Khon Sothea", , , 3
End Sub
```

.....

ទំរង់នៃ Obj Collection Object ដែលរក្សាទិន្នន័យគឺ :

	Obj Object
1	Srey Phean
2	Koub Sopheak
3	To SokNang
4	<i>Khon Sothea</i>
5	Kim SaoRin

ឧទាហរណ៍ទី៤ :

☞ Source Code:
 Option Explicit
 Dim Obj As New Collection

.....

```
Private Sub Form_Load( )
  Obj.Add "Srey Phean", "k1"
  Obj.Add "Koub Sopheak", "k2"
  Obj.Add "To SokNang", "k3"
  Obj.Add "Kim SaoRin"
  Obj.Add "Khon Sothea"
End Sub
```

.....

ទំរង់នៃ Obj Collection Object ដែលរក្សាទិន្នន័យគឺ :

	Obj Object	Key តំណាងដោយធាតុនៃ Collection
1	Srey Phean	K1
2	Koub Sopheak	K2
3	To SokNang	K3
4	Kim SaoRin	
5	Khon Sothea	

ឧទាហរណ៍ទី៥ :

☞ Source Code:
 Option Explicit
 Dim Obj As New Collection

.....

```
Private Sub Form_Load( )
```

```

Obj.Add "Srey Phean", "k1"
Obj.Add "Koub Sopheak", "k2"
Obj.Add "To SokNang", "k3"
Obj.Add "Kim SaoRin"
Obj.Add "Khon Sothea", , "k1"
End Sub

```

ទំរង់នៃ Obj Collection Object ដែលរក្សាទិន្នន័យគឺ :

	Obj Object	Key កំណាងអោយទិន្នន័យ
1	Khon Sothea	
2	Srey Phean	K1
3	Koub Sopheak	K2
4	To SokNang	K3
5	Kim SaoRin	

II.3. ការប្រើ Count Method ដើម្បីរាប់ចំនួន Item ដែលមាននៅក្នុង Collection's Object

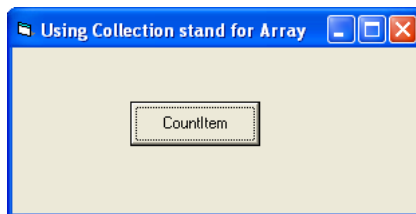
Syntax:

```
Object.Count
```

ឧទាហរណ៍ :

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.៨។

រូបភាពទី៦.៨



☞ Source Code:

```

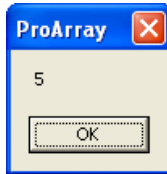
Option Explicit
Dim Obj As New Collection

```

```
Private Sub Form_Load()
    Obj.Add "Srey Phean", "k1"
    Obj.Add "Koub Sopheak", "k2"
    Obj.Add "To SokNang", "k3"
    Obj.Add "Kim SaoRin"
    Obj.Add "Khon Sothea", , "k1"
End Sub
```

```
Private Sub cmdCount_Click()
    MsgBox Obj.Count
End Sub
```

លទ្ធផល :



II.4. ការអានទិន្នន័យ (Item) ចេញពី Collection's Object

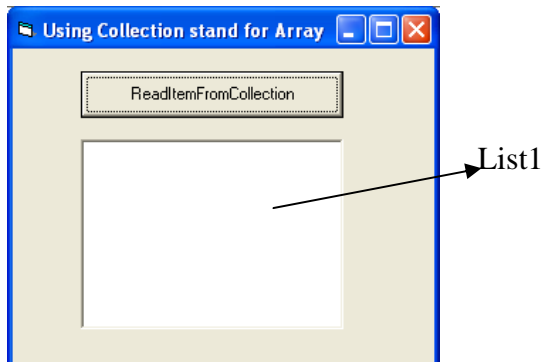
Syntax:

```
Object.Item(Index)
រឺ
Object.Item(Key)
```

ឧទាហរណ៍ទី១ :

នៅក្នុងឧទាហរណ៍នេះយើងនឹងធ្វើការអានទិន្នន័យទៅតាម Index របស់ Item នីមួយៗ។
ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.៩។

រូបភាពទី៦.៩



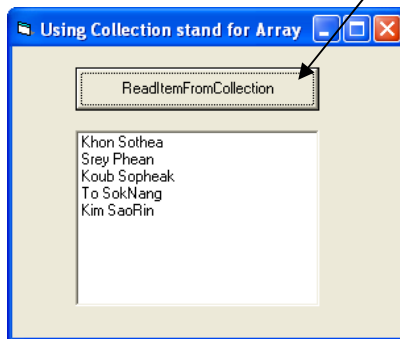
☞ Source Code:

```
Option Explicit  
Dim Obj As New Collection
```

```
Private Sub Form_Load()  
    Obj.Add "Srey Phean", "k1"  
    Obj.Add "Koub Sopheak", "k2"  
    Obj.Add "To SokNang", "k3"  
    Obj.Add "Kim SaoRin"  
    Obj.Add "Khon Sothea", , "k1"  
End Sub
```

```
Private Sub cmdReadItemFromCollection_Click()  
    Dim i%  
    For i = 1 To Obj.Count  
        Me.List1.AddItem Obj.Item(i)  
    Next  
End Sub
```

លទ្ធផល :



ឧទាហរណ៍ទី២ :

នៅក្នុងឧទាហរណ៍នេះយើងនឹងធ្វើការអានទិន្នន័យទៅតាម Key របស់ Item នីមួយៗ វិញម្តង។
ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.៩។

☞ Source Code:

```
Option Explicit  
Dim Obj As New Collection
```

```

Private Sub Form_Load( )
    Obj.Add "Srey Phean", "k1"
    Obj.Add "Koub Sopheak", "k2"
    Obj.Add "To SokNang", "k3"
    Obj.Add "Kim SaoRin"
    Obj.Add "Khon Sothea", , "k1"
End Sub

```

```

Private Sub cmdReadItemFromCollection_Click()
    Dim i%
    Me.List1.AddItem Obj.Item("k1")
    Me.List1.AddItem Obj.Item("k2")
    Me.List1.AddItem Obj.Item("k3")
End Sub

```

លទ្ធផល :



II.5. ការលុបធាតុនៃយ (Item) យចេញពី Collection's Object

Syntax:

```

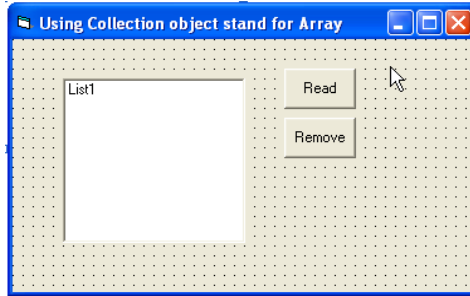
Object.Remove(Index)
រឺ
Object.Remove(Key)

```

ឧទាហរណ៍ទី១ :

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពទី៦.១១។

រូបភាពទី៦.១១



☞ Source Code:

```
Option Explicit  
Dim Obj As New Collection
```

```
Private Sub Form_Load()  
    Obj.Add "Srey Phean", "k1"  
    Obj.Add "Koub Sopheak", "k2"  
    Obj.Add "To SokNang", "k3"  
    Obj.Add "Kim SaoRin"  
    Obj.Add "Khon Sothea", , "k1"  
End Sub
```

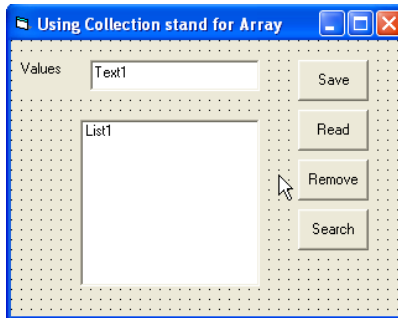
```
Private Sub cmdRead_Click()  
    Dim i%  
    Me.List1.Clear  
    For i = 1 To Obj.Count  
        Me.List1.AddItem Obj.Item(i)  
    Next  
End Sub
```

```
Private Sub cmdRemove_Click()  
    On Error GoTo Err_NoKeyForItem  
    Obj.Remove "k1"  
    Obj.Remove "k2"  
    Exit Sub  
Err_NoKeyForItem:  
    MsgBox "No key represent the item to remove", vbInformation  
End Sub
```

ឧទាហរណ៍ទី២ :

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី៦.១០ ។

រូបភាពទី៦.១០



គោលបំណង :

- ពេលដែលលោកអ្នកចុចលើ Save Button សូមបញ្ចូលទិន្នន័យពី Text1 ចូលទៅក្នុង Collection Object ។
- ពេលដែលលោកអ្នកលើ Read Button សូមអានទិន្នន័យចេញពី Collection Object មកដាក់លើ List1 ។
- ពេលដែលចុចលើ Remove Button សូមអោយលុបទៅតាម InputBox ដែលលោកអ្នកវាយបញ្ចូល។
- ពេលដែលចុចលើ Search Button សូមអោយរុករក Item ទៅតាម InputBox ដែលលោកអ្នកវាយបញ្ចូល។

Source Code:

```
Option Explicit
Dim Obj As New Collection
```

```
.....
Private Sub Form_Load()
    Me.Text1 = ""
End Sub
```

```
.....
Private Sub cmdSave_Click()
    Obj.Add Me.Text1.Text
    Me.Text1 = ""
    Me.Text1.SetFocus
End Sub
```

```
Private Sub cmdRead_Click()  
    Call ReadItemFromCollection(Obj, Me.List1)  
End Sub
```

```
Private Sub cmdRemove_Click()  
    Dim st As String  
    st = InputBox("Please, enter Item do you want to delete?")  
    If st = "" Then  
        MsgBox "You must enter Item. try again!!"  
        Exit Sub  
    Else  
        If CheckItem(Obj, st) = True Then  
            Call RemoveItemFromCollection(Obj, st)  
        Else  
            MsgBox "Item Not Found"  
        End If  
    End If  
    Call ReadItemFromCollection(Obj, Me.List1)  
End Sub
```

```
Private Sub cmdSearch_Click()  
    Dim st As String  
    Dim i%  
    st = InputBox("Please, enter Item do you want to delete?")  
    If st = "" Then  
        MsgBox "You must enter Item. try again!!"  
        Exit Sub  
    Else  
        If CheckItem(Obj, st) = True Then  
            Me.List1.Clear  
            For i = 1 To Obj.Count  
                If StrComp(Obj.item(i), st, vbTextCompare) = 0 Then  
                    Me.List1.AddItem Obj.item(i)  
                End If  
            Next  
        Else  
            MsgBox "Item Not Found"  
        End If  
    End If  
End Sub
```

```
Sub RemoveItemFromCollection(ObjVar As Collection, item As Variant)  
    Dim i%
```

```

For i = ObjVar.Count To 1 Step -1
    If StrComp(ObjVar.item(i), item, vbTextCompare) = 0 Then
        ObjVar.Remove i
    End If
Next
End Sub

```

```

.....
Function CheckItem(ObjVar As Collection, item As Variant) As Boolean
    Dim i%
    For i = 1 To ObjVar.Count
        If StrComp(ObjVar.item(i), item, vbTextCompare) = 0 Then
            CheckItem = True
            Exit Function
        End If
    Next
End Function

```

```

.....
Sub ReadItemFromCollection(ObjVar As Collection, Lst As ListBox)
    Dim i%
    Lst.Clear
    For i = 1 To ObjVar.Count
        Lst.AddItem ObjVar.item(i)
    Next
End Sub

```

III. ការប្រើ For each ... Next ជាមួយ Collection's Object

Syntax:

```

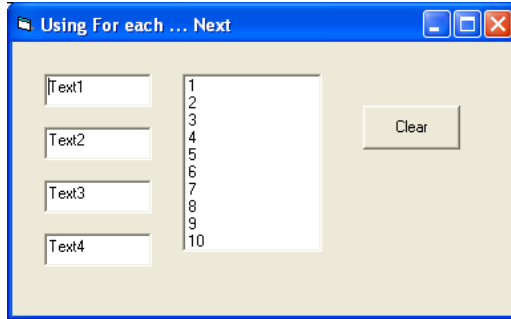
For each ObjectName In Collection's Object
    Loop Statement
Next

```

ឧទាហរណ៍ :

ឧបមាថាលោកអ្នកមានទំរង់ Form ដូចរូបភាពទី៦.១២។

រូបភាពទី៦.១១



គោលបំណង :

ពេលដែលលោកអ្នកចុចលើ Clear Button សូមសំអាតរាល់ទិន្នន័យទាំងអស់ ដែលមាននៅក្នុង Text Box និង List Box ។

☞ Source Code:

Option Explicit

```
.....
Private Sub Form_Load()
    Dim i%
    For i = 1 To 10
        Me.List1.AddItem i
    Next
End Sub
```

```
.....
Private Sub cmdClear_Click()
    Call Clear(Me)
End Sub
```

```
.....
Sub Clear(frm As Form)
    Dim Ctr As Control
    For Each Ctr In frm.Controls
        If TypeOf Ctr Is TextBox Then
            Ctr.Text = ""
        ElseIf TypeOf Ctr Is ListBox Then
            Ctr.Clear
        End If
    Next
End Sub
.....
```

~~វិភាគលើ~~ Clear Sub Procedure :

-ចំពោះ Dim Ctr As Control មានន័យថាយើងធ្វើការប្រកាសអថេរមួយដែល មានប្រភេទជា Control Object (ដូចជា TextBox, ListBox, ComboBox, CommandButton, . . .) ។

- ចំពោះ Controls ជា Control ទាំងអស់។

- ចំពោះ Code : For Each Ctr In frm.Controls គឺធ្វើការ Scan រាល់ Control ទាំងអស់ដែលមាននៅលើ Form (ដែលយើងបានប្រើអថេរជាប្រភេទ Form គឺ Frm ដែលតំណាងអោយ Form ដែលលោកអ្នកកំពុងធ្វើការលើវា) ។

-If TypeOf Ctr Is TextBox then មានន័យថាយើងធ្វើការសិក្សាថាតើ Ctr ដែលវា Scan ឃើញនោះជា TextBox ដែរឬទេ ។ ប្រសិនបើវាជា TextBox នោះវានឹងធ្វើការសំអាតរាល់ទិន្នន័យក្នុង TextBox ។

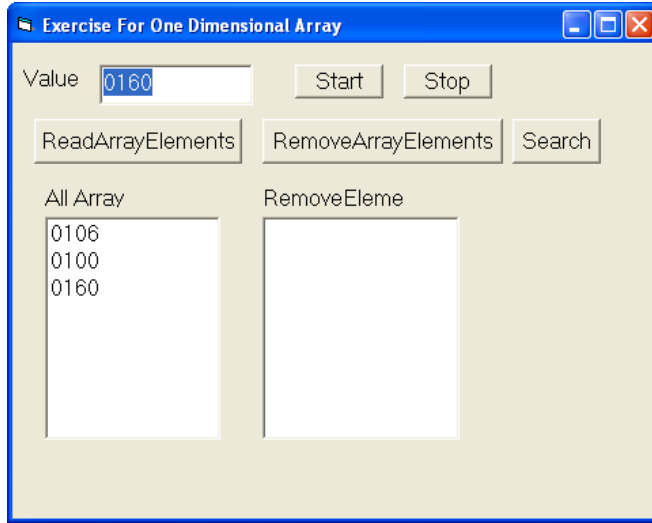
- ElseIf TypeOf Ctr Is ListBox Then មានន័យថាយើងធ្វើការសិក្សាថាតើ Ctr ដែលវា Scan ឃើញនោះជា ListBox ដែលឬទេ ។ ប្រសិនបើវាជា ListBox នោះវានឹងធ្វើការសំអាតរាល់ទិន្នន័យក្នុង ListBox ។

===== *

លំហាត់អនុវត្ត

1. លំហាត់ទី១

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពខាងក្រោម :



គោលបំណង :

- ពេលចុចទៅលើ Start Button ត្រូវធ្វើអោយ Text1 បង្វិលលេខដែលបិតនៅ ចន្លោះពី 0 ដល់ ១០០០ ហើយអោយចេញចំនួន៤ Digits ។
- ពេលចុចទៅលើ Button Stop អោយវាលប់ហើយ បញ្ចូលតំលៃនោះទៅក្នុង Array មួយ។
- ពេលចុច Button Read សូមអាន Array Elements ទាំងអស់មកបញ្ចូលក្នុង List Box ខាងឆ្វេង។
- ពេលចុច Button Remove សូម Select យក Item ណាមួយដែលមាននៅក្នុង List ខាងឆ្វេង ជាមុនសិន ពេល Select រួចហើយចុច Button Remove សូមអោយលុប Element និង Index របស់ Array ដែលត្រូវគ្នានឹង Item របស់ List ខាងឆ្វេង ហើយ យក Element ដែលលុបចេញនោះមកបញ្ចូលក្នុង List Box ខាងស្តាំ។
- ពេលចុច Button Search សូម Select យក Item ណាមួយដែលមាននៅក្នុង List ខាងឆ្វេង ឬ List Box ខាងស្តាំជាមុនសិន ហើយបើ Item នោះមាននៅក្នុង Array សូមបង្ហាញ Message Found ពុំនោះទេ សូមបង្ហាញ Message Not Found ។

ចំណាំ :

- ដើម្បីលុបរាល់ Elements របស់ Array ទាំងអស់សូមប្រើ Erase Keyword ។
Syntax:

```
Erase ArrayName
```

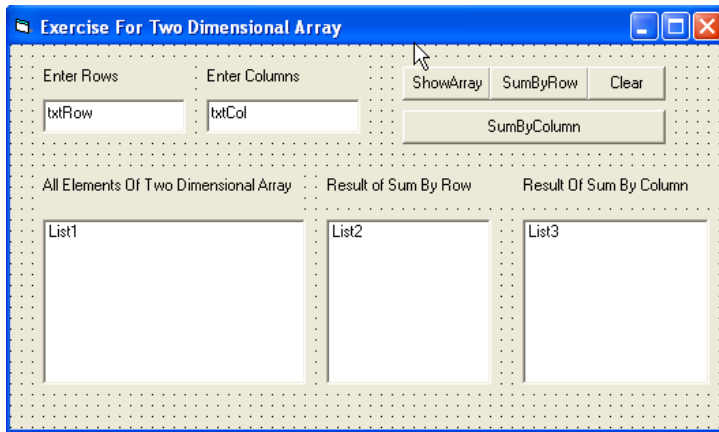
- លោកអ្នកអាចបញ្ជូនតំលៃរាល់ elements របស់ Array ទាំងអស់ទៅអោយ Array មួយបានតាមរយៈ Syntax ខាងក្រោម ។

Syntax:

```
ArrayName1=ArrayName2
```

2. លំហាត់ទី២

ឧបមាថាលោកអ្នកមានទម្រង់ Form ដូចរូបភាពខាងក្រោម :



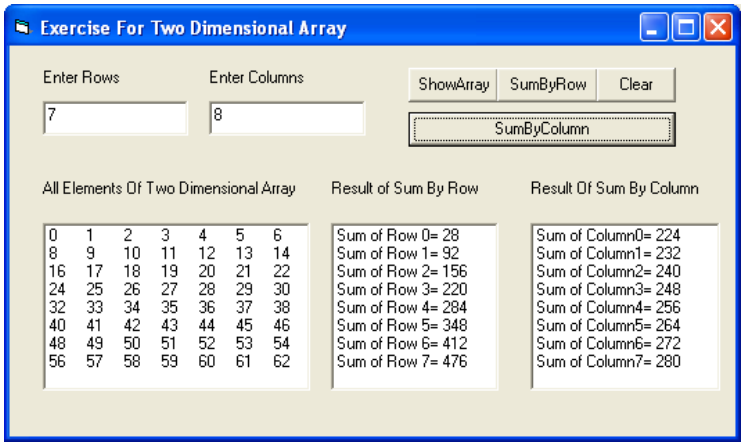
គោលបំណង :

- សូមបញ្ចូល ចំនួន Rows និង Columns ត្រង់ txtRow Text Box ERows និង txtCol TextBox ។
- ពេលចុច Show Array Button សូមអោយបញ្ចូលទិន្នន័យទៅអោយធាតុ របស់ Array ដោយអោយធាតុរបស់វាកើនទៅតាមលំដាប់ស្វ៊ីតដែលមានតួទី១ស្មើ 0 ហើយ ផលសងរួមរបស់វាក៏ ១ បន្ទាប់មកបញ្ជាញតំលៃធាតុរបស់ Array ទាំងអស់នៅក្នុង List1 List Box ។

- ពេលចុច SumByColumn Button សូមធ្វើការគណនាទៅលើផលបូកជួរឈរ នីមួយៗ ហើយបង្ហាញវានៅក្នុង List3 ListBox ។

- ពេលចុច SumByRow Button សូមធ្វើការគណនាទៅលើផលបូកនៃជួរដេក នីមួយៗ ហើយបង្ហាញវានៅក្នុង List2 ListBox ។

- ចំពោះ txtRow និង txtCol អនុញ្ញាតអោយវាយបញ្ចូលបានតែលេខ ។
សូមមើលទម្រង់លទ្ធផលពេលដំណើរការដូចខាងក្រោម :



3. លំហាត់ទី៣

សូមធ្វើលំហាត់ទី១ឡើងវិញដោយផ្លាស់ប្តូរពីការប្រើ Array មកប្រើជា Collection វិញ។

កំណែលើលំហាត់ទី១ :

```
Source Code:
Option Explicit
Dim B As Boolean
Dim ArrValue() As String
Dim Indx As Integer
```

```
Private Sub Form_Load()
    B = True
End Sub
```

```

Private Sub Timer1_Timer()
    Me.txtValue = Format(Rnd() * 1000, "0000")
    If B = False Then
        ReDim Preserve ArrValue(Indx)
        ArrValue(Indx) = Me.txtValue
        Indx = Indx + 1
    End If
End Sub

```

```

Private Sub CmdStart_Click()
    Me.Timer1.Interval = 100
    B = True
End Sub

```

```

Private Sub CmdStop_Click()
    B = False
    Me.Timer1.Interval = 0
End Sub

```

```

Private Sub CmdRead_Click()
    Call readArrValue(ArrValue, Me.LstAll)
End Sub

```

```

Private Sub CmdRemove_Click()
    Dim Tem() As String
    If Me.LstAll.ListIndex <> -1 Then
        'A = DeleteArrElements(ArrValue, Me.LstAll.ListIndex)
        Call DeteleArrElements(ArrValue, Me.LstAll.List(Me.LstAll.ListIndex))
    Else
        MsgBox "Please,select one item data on list"
    End If
End Sub

```

```

Private Sub CmdSearch_Click()
    If Me.LstAll.ListIndex <> -1 Then
        If CheckValue(ArrValue, Me.LstAll.List(Me.LstAll.ListIndex)) = True
            Then
                MsgBox "Found Value of this Array Element"
            Else
                MsgBox "Not found"
            End If
    End If
End Sub

```



```

    End If
Else
    MsgBox "Please, select one item on list Box"
End If
End Sub

```

```

Sub readArrValue(Arr() As String, Lstbox As ListBox)
    Lstbox.Clear
    On Error GoTo SubScriptOutOfRang
    For i = LBound(Arr) To UBound(Arr)
        Lstbox.AddItem Arr(i)
    Next
SubScriptOutOfRang:
    Exit Sub
End Sub

```

```

Function CheckValue(Arr() As String, Value As String) As Boolean
    For i = LBound(Arr) To UBound(Arr)
        If StrComp(Arr(i), Value, vbTextCompare) = 0 Then
            CheckValue = True
            Exit Function
        End If
    Next
End Function


```

```

Sub DeteleArrElements(Arr() As String, Value As String)
    Dim TempArr() As String
    For i = LBound(Arr) To UBound(Arr)
        If Arr(i) <> Value Then
            ReDim Preserve TempArr(j)
            TempArr(j) = Arr(i)
            j = j + 1
        Else
            Indx = Indx - 1
            'ReDim Preserve Arr(Indx - 1)
        End If
    Next
    Erase Arr
    Arr = TempArr
End Sub

```

កំណែលើបំណាត់ទី២ :

 Source Code:

```
Option Explicit  
Dim ArrValue() As Long
```

```
Private Sub Form_Activate()  
    Clear Me  
End Sub
```

```
Private Sub cmdClear_Click()  
    Clear Me  
    Me.txtRow.SetFocus  
End Sub
```

```
Private Sub CmdShowArray_Click()  
    If Me.txtCol = "" Then  
        MsgBox "Please, enter Column Count"  
        Me.txtCol.SetFocus  
        Exit Sub  
    ElseIf Me.txtRow = "" Then  
        MsgBox "Please, enter Row Count"  
        Me.txtRow.SetFocus  
        Exit Sub  
    Else  
        ReDim ArrValue(CLng(Me.txtRow), CLng(Me.txtRow))  
        Call ShowArrayElements(Me.List1, ArrValue)  
    End If  
End Sub
```

```
Private Sub cmdSumByCol_Click()  
    Call SumByColumn(Me.List3, ArrValue)  
End Sub
```

```
Private Sub cmdSumByRow_Click()  
    Call SumByRow(Me.List2, ArrValue)  
End Sub
```

```
Sub ShowArrayElements(Lst As ListBox, A() As Long)  
    Dim i%, j%, Number!
```

```

Dim st As String
For i = LBound(A, 1) To UBound(A, 1)
    For j = LBound(A, 2) To UBound(A, 2)
        A(i, j) = Number
        Number = Number + 1
    Next
Next

For i = LBound(A, 1) To UBound(A, 1)
    For j = LBound(A, 2) To UBound(A, 2)
        Select Case Len(Str(A(i, j)))
            Case 1
                st = st & A(i, j) & Space(9)
            Case 2
                st = st & A(i, j) & Space(7)
            Case 3
                st = st & A(i, j) & Space(5)
        End Select
    Next
    Lst.AddItem st
    st = ""
Next
End Sub

```

```

Private Sub txtCol_KeyPress(KeyAscii As Integer)
    If ControlKey("1234567890" & Chr(13) & Chr(8), Chr(KeyAscii)) = False
        Then
            MsgBox "Invalid number"
            KeyAscii = 0
        End If
    End Sub

```

```

Private Sub txtRow_KeyPress(KeyAscii As Integer)
    If ControlKey("1234567890" & Chr(13) & Chr(8), Chr(KeyAscii)) = False
        Then
            MsgBox "Invalid number"
            KeyAscii = 0
        End If
    End Sub

```

```

Sub Clear(frm As Form)
    Dim ctr As Control
    For Each ctr In frm.Controls

```

```
    If TypeOf ctr Is TextBox Then
        ctr.Text = ""
    ElseIf TypeOf ctr Is ListBox Then
        ctr.Clear
    End If
Next
End Sub
```

```
Sub SumByRow(Lst As ListBox, A() As Long)
    Dim Row%, Col%
    Dim SumRow!
    Lst.Clear
    For Row = LBound(A, 1) To UBound(A, 1)
        For Col = LBound(A, 2) To UBound(A, 2)
            SumRow = SumRow + A(Row, Col)
        Next
        Lst.AddItem "Sum of Row " & Row & "= " & SumRow
        SumRow = 0
    Next
End Sub
```

```
Sub SumByColumn(Lst As ListBox, A() As Long)
    Dim Row%, Col%
    Dim SumCol!
    Lst.Clear
    For Col = LBound(A, 1) To UBound(A, 1)
        For Row = LBound(A, 2) To UBound(A, 2)
            SumCol = SumCol + A(Row, Col)
        Next
        Lst.AddItem "Sum of Column" & Col & "= " & SumCol
        SumCol = 0
    Next
End Sub
```

```
Function ControlKey(Source As String, SearchString As String) As Boolean
    If InStr(1, Source, SearchString, vbTextCompare) <> 0 Then
        ControlKey = True
        Exit Function
    End If
End Function
```

 កំណែទើបបំបាត់ទី៣ :

សូមរៀនធ្វើ លំហាត់ទី៣ដោយខ្លួនឯង។

===== *