

Arrays: *part2*

Multi-dimensional Arrays

It's called also (**Two- dimensional Array**), it can have more than one dimension to represent data as a table. A two dimensional array takes the **row-column** form.

For example:

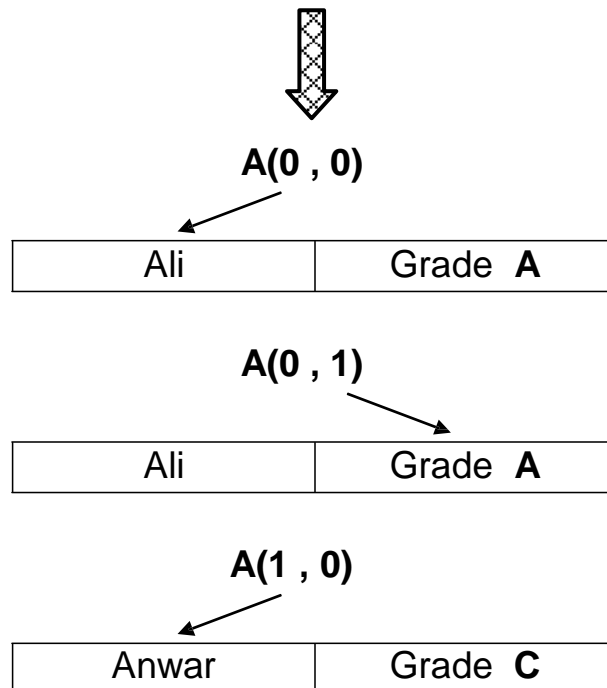
Dim A(5, 5) As Integer “ it's **6-by-6** array because index starts from **0**”

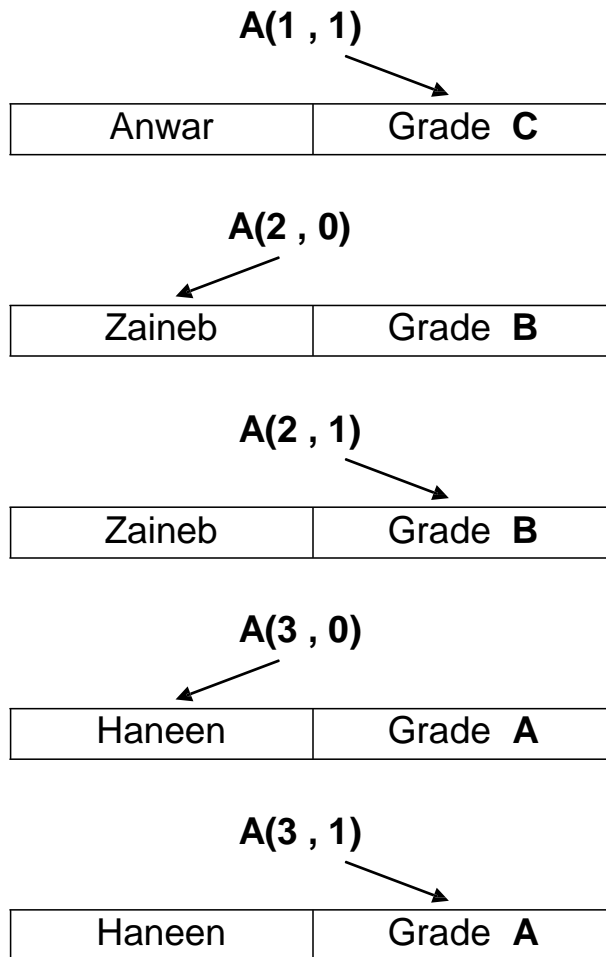
Another form:

Dim value(1 To 4, 1 To 3) As string “it's **4** rows and **3** columns”

Explain:

Dim A(3,1) As String “it's **4** rows and **2** columns”





Example1: Write a **4 – by – 2** array contains student names and grade.

Solution:

Code:

Dim A(3, 1) As String

Command1:

Private Sub Command1_Click()

Form1.Cls

A(0, 0) = "Ali"

A(0, 1) = "A"

Student Name	Grade
Ali	A
Anwar	C
Zaineb	B
Haneen	A

OK

A(1, 0) = "Anwar"

A(1, 1) = "C"

A(2, 0) = "Zaineb"

A(2, 1) = "B"

A(3, 0) = "Haneen"

A(3, 1) = "A"

Print " " ; "Student " ; " " ; "Grade"

For i = 0 To 3

For j = 0 To 1

Print " " ; (A(i, j)) ; " " ,

Next j

Print

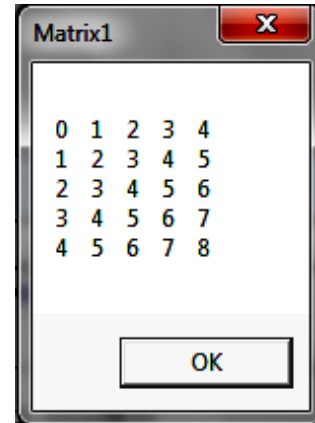
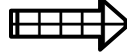
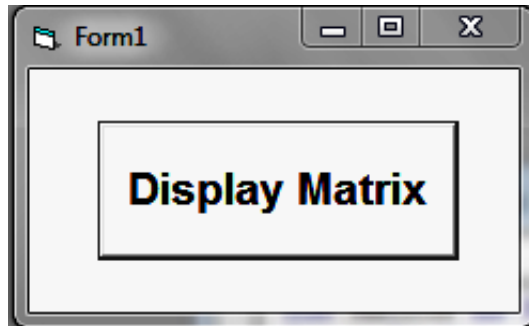
Next i

End Sub

Example2 : Write a **5-by-5** Array to display a Matrix numbers showing that each row and each column will increase by **1** number respectively as in the figure:

Solution:

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8



Code:

```
Dim x(4, 4) As Integer
Dim Matrix As String
Dim i As Integer
Dim j As Integer
```

Command1:

```
Private Sub Command1_Click()
```

Reading {

```
For i = 0 To 4
For j = 0 To 4
x(i, j) = i + j
Next j
Next i
```

لجمع العنصر مع العنصر الذي يليه حتى يتم تحقق شرط السؤال

```
Matrix = " "
```

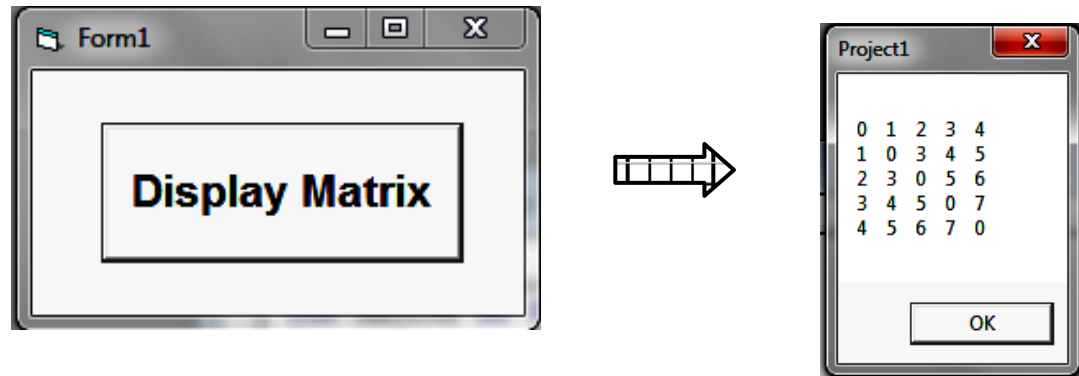
Printing {

```
For i = 0 To 4
For j = 0 To 4
Matrix = Matrix & x(i, j) & " "
Next j
Matrix = Matrix & Chr(13) & Chr(10)
Next i
MsgBox Matrix, vbDefaultButton1 + vbOKOnly
```

```
End Sub
```

Example3 : Write a **5-by-5** Array to display a Matrix numbers showing that each element in the main diagonal is equal to **0**

Solution:



Code:

```
Dim x(4, 4) As Integer
Dim Matrix As String
Dim i As Integer
Dim j As Integer
```

Command1:

```
Private Sub Command1_Click()
```

Reading {

```
For i = 0 To 4
For j = 0 To 4
x(i, j) = i + j
Next j
Next i
```

Loop for diagonal {

```
For i = 0 To 4
For j = 0 To 4
If i = j Then
x(i, j) = 0 End
If
Next j
Next i
```

When (i) is equal to (j) means we are deal with the main diagonal

Matrix = " "

Printing {

```

For i = 0 To 4
For j = 0 To 4
Matrix = Matrix & x(i, j) & " "
Next j
Matrix = Matrix & Chr(13) & Chr(10)
Next i
MsgBox Matrix, vbDefaultButton1 + vbOKOnly

```

End Sub

Example4 : Write a **10-by-10** Array to display a Matrix numbers each element equal to **1**

Solution:

Code:

```

Dim Matrix(9, 9) As Integer
Dim Row, Column As Integer

```

Command1:

```

Private Sub Command1_Click()

```

```

Form1.Cls

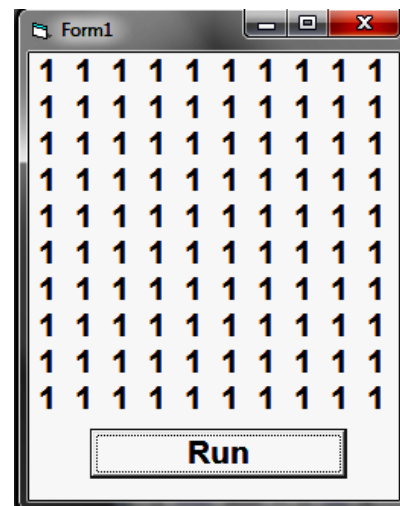
```

Reading {

```

For Row = 0 To 9
For Column = 0 To 9
Matrix(Row, Column) = 1
Next Column
Next Row

```



```

Printing {
For Row = 0 To 9
For Column = 0 To 9
Print Matrix(Row, Column);
Next Column
Print
Next Row

End Sub

```

Example5 : Write a VB6 program to make **Addition** of two matrices, each matrix consists of **2-by-2** size. Display the result on the **form**.

Solution:

Code:

Dim Matrix1(1, 1) As Integer, Matrix2(1, 1), Sum(1, 1) As Integer

Command1:

```

Private Sub cmdSum_Click()
Form1.Cls

```

```

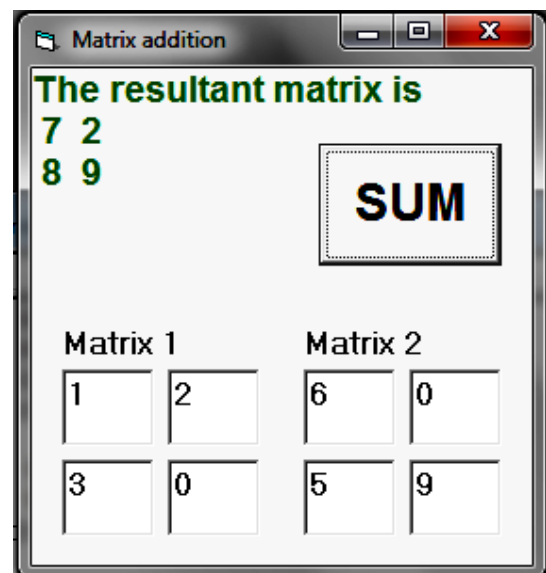
Matrix1(0, 0) = Val(Text1.Text)
Matrix1(0, 1) = Val(Text2.Text)
Matrix1(1, 0) = Val(Text3.Text)
Matrix1(1, 1) = Val(Text4.Text)

```

```

Matrix2(0, 0) = Val(Text5.Text)
Matrix2(0, 1) = Val(Text6.Text)
Matrix2(1, 0) = Val(Text7.Text)
Matrix2(1, 1) = Val(Text8.Text)

```



Reading {
For i = 0 To 1
For j = 0 To 1
Sum(i, j) = Matrix1(i, j) + Matrix2(i, j)
Next j
Next i

Printing {
Print "The resultant matrix is:"
For i = 0 To 1
For j = 0 To 1
Print Sum(i, j);
Next j
Print " "
Next i

End Sub

H.W

- 1) Write a **5-by-5** Array to display a Matrix numbers showing that each element in the upper triangle is equal to **0**