

# مبادئ علوم الحاسب 2



اعداد

د/ سهام عليو عامر  
مدرس علوم الحاسب  
كلية الحاسبات والمعلومات جامعة سوهاج

تابع ... الباب الثاني \_3

لغة فيجوال بيزك دوت نت

VB.net

Microsoft

Visual Basic



# المصفوفات Arrays

▶ طبقا للمتغيرات التي سبق دراستها, فمثلا اذا اردنا حساب معدل طالب معين في عشرة مواد يدرسها فاننا سوف نعلن عن عشر متغيرات رقمية لاستقبال نقاط الطالب بالطريقة التالية:

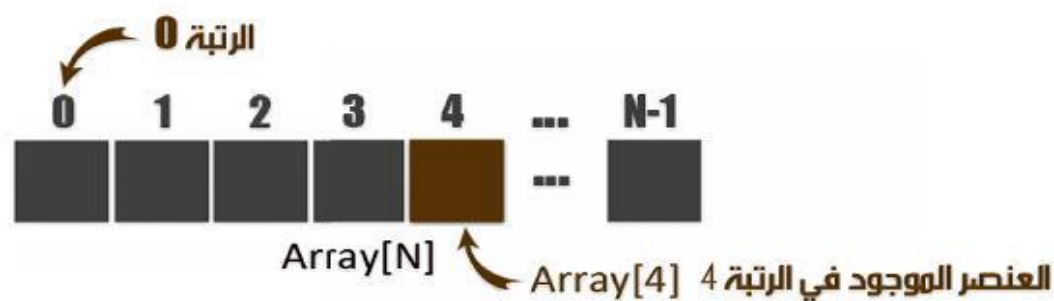
```
Dim Value1, Value2, Value3, Value4, Value5, Value6,  
Value7, Value8, Value9, Value10 As Double
```

▶ لكن هذه الطريقة ليست مجدية اذا كنا نحتاج الى تخزين قيم كثيرة. لذلك نستخدم المصفوفات لتخزين مجموعة من القيم المنتمية لنفس نوع البيانات في متغير واحد, ويرتكز مفهوم المصفوفات على ركيزتين اساسيتين هما:

▶ القيمة Value : وهي القيم المراد تخزينها في عناصر المصفوفة.

▶ الرتبة Index : وهي رتبة العنصر داخل المصفوفة, وتبدأ بصفر وتنتهي برتبة اخر عنصر ناقص واحد.

▶ نستطيع الوصول الى اي عنصر من عناصر المصفوفة من خلال رتبته Index  
ويسمى هذا النوع من المصفوفات بالمصفوفات احادية لبعده - One  
Dimensional Array .



للإعلان عن مصفوفة أحادية في لغة الفيجوال بيسك،

```
Dim Marks(9) As Double
```

▶ المصفوفة اعلاه مكونة من عشرة عناصر عشرية لان الترتيب يبدأ من صفر وينتهي  
باخر رتبة في المصفوفة.

## مثال:

```
Dim Marks() As Double = {20, 19, 14, 17, 13, 11.5,  
16.25, 18.75, 9, 15}
```

رتبة العنصر	قيمه
0	20
1	19
2	14
3	17
4	13
5	11.5
6	16.25
7	18.75
8	9
9	15

- ويمكننا كذلك اسناد القيم بعد الاعلان عن المصفوفة وتحديد عدد عناصرها كما يوضح المثال التالي الذي يؤدي نفس الكود السابق:

`Dim Marks(9) As Double`

`Marks(0) = 20`

`Marks(1) = 19`

`Marks(2) = 14`

`Marks(3) = 17`

`Marks(4) = 13`

`Marks(5) = 11.5`

`Marks(6) = 16.25`

`Marks(7) = 18.75`

`Marks(8) = 9`

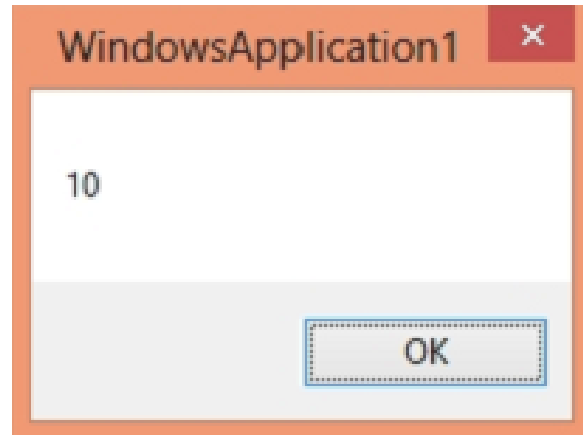
`Marks(9) = 15`

## الحصول على طول المصفوفة:

يمكننا معرفة عدد العناصر المكونة للمصفوفة باستخدام الخاصية Length كما في المثال التالي:

```
Dim Marks() As Double = {20, 19, 14, 17, 13, 11.5,  
16.25, 18.75, 9, 15}  
MsgBox(Marks.Length)
```

▶ بعد تنفيذ الكود سنحصل على رسالة مفادها ان عدد العناصر المكونة للمصفوفة هو 10 عناصر كالتالي:



► ويمكننا ايضا الحصول على طول المصفوفة باستخدام الخاصية Count كما يبين المثال التالي:

```
Dim Marks() As Double = {20, 19, 14, 17, 13, 11.5,  
16.25, 18.75, 9, 15}
```

```
MsgBox(Marks.Count)
```

سنحصل على نفس النتيجة السابقة.

► لعرض عنصر من عناصر المصفوفة او استخدامه في عملية معينة نحدد رتبة العنصر الذي نريد كما تعرض الامثلة الاتية:

```
Dim Marks() As Double = {20, 19, 14, 17, 13, 11.5,  
16.25, 18.75, 9, 15}
```

```
Dim Value As Double = Marks(2)
```

```
MsgBox(Marks(8))
```

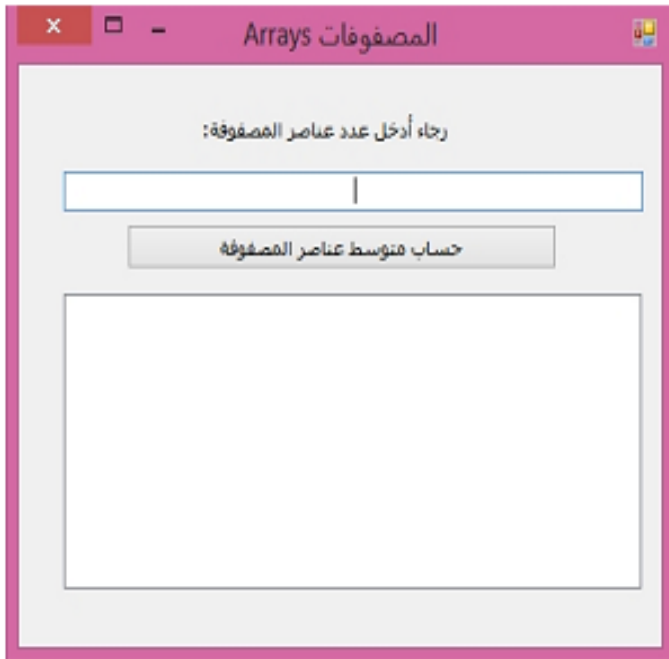
```
Dim Sum As Double = Marks(1) + Marks(4)
```



## مثال:

برنامج بسيط يطلب من المستخدم ان يدخل عدد عناصر مصفوفة رقمية ثم نقوم باسناد قيم عشوائية لهذه العناصر ثم نحسب المتوسط Average ونعرضه في رسالة ونطبع قيم العناصر في اداة ListBox.

قم بانشاء مشروع جديد من نوع Windows Form Application وضع عليه الادوات التالية:



الأداة	اسمها	دورها
Label	lblArray	من أجل عرض السؤال
TextBox	txtArray	من أجل إدخال عدد عناصر المصفوفة
Button	btnAvg	من أجل حساب متوسط المصفوفة
ListBox	lbDetails	من أجل عرض قيم عناصر المصفوفة

▶ الان انقر مرتين على الزر btnAvg للولوج للحدث Click وقم بكتابة الكود التالي:

```
Dim ItemCount As Integer = Val(txtArray.Text)

Dim intArray(ItemCount) As Integer

Dim Rand As New Random

Dim Sum As Integer = 0

For Count As Integer = 0 To intArray.Length - 1

    intArray(Count) = Rand.Next(100)

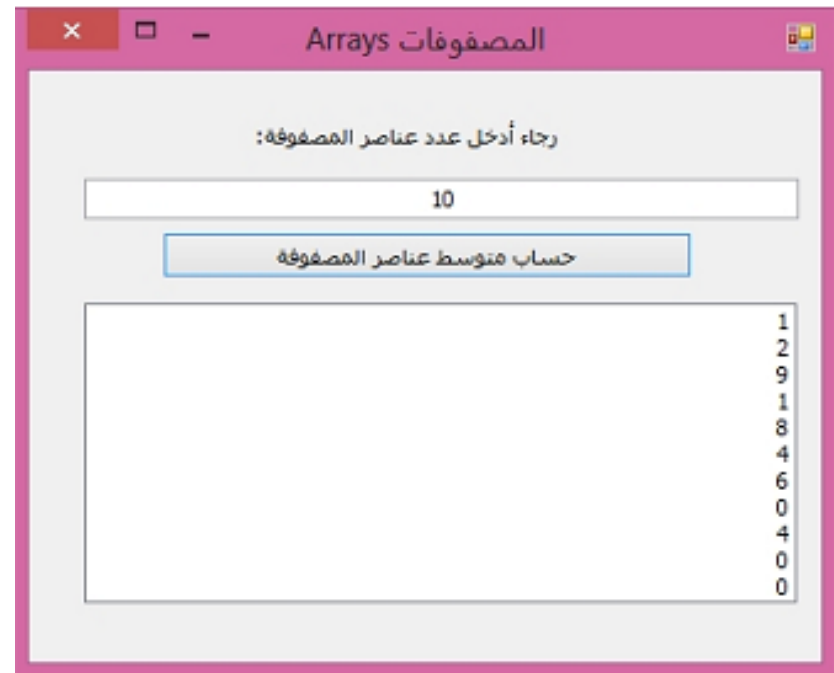
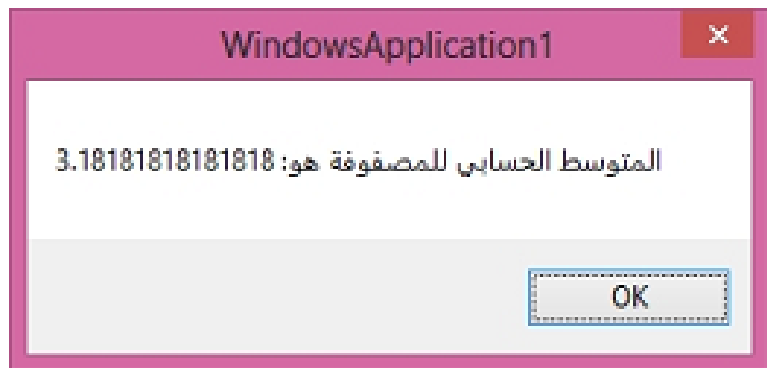
    lbDetails.Items.Add(intArray(Count))

    Sum += intArray(Count)
Next

Dim Average As Double = Sum / intArray.Length

MsgBox("المتوسط الحسابي للمصفوفة هو: " & Average)
```

- لو نفذنا الكود السابق وادخلنا مثلا القيمة 10 كعدد لعناصر المصفوفة فسوف تكون النتيجة كما يلي ( مع الاخذ في الاعتبار ان قيم العناصر ستكون عشوائية):



## مثال:

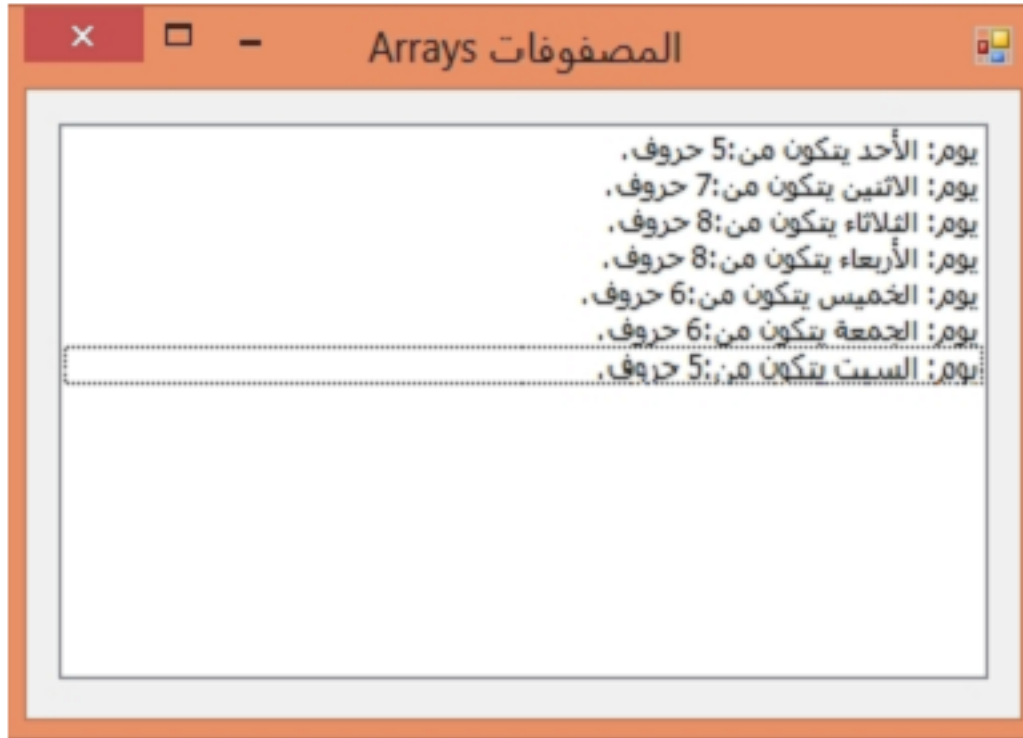
▶ البرنامج الذي سنقوم بإنشائه الآن هو برنامج بسيط يقوم بعرض قيمة كل عنصر مصحوبة بعدد الحروف المكونة له:

```
Dim WeekArray(6) As String
```

```
WeekArray(0) = "الأحد"  
WeekArray(1) = "الاثنين"  
WeekArray(2) = "الثلاثاء"  
WeekArray(3) = "الأربعاء"  
WeekArray(4) = "الخميس"  
WeekArray(5) = "الجمعة"  
WeekArray(6) = "السبت"
```

```
For Each Day As String In WeekArray  
    lbDetails.Items.Add("من يتكون " & Day & " يوم: " &  
Day.Length & " حروف. ")  
Next
```

بعد تنفيذ الكود السابق ستكون النتيجة هكذا:



- ويمكننا أيضا ترتيب عناصر المصفوفة النصية ترتيبا ابجديا والمصفوفة الرقمية من الاصغر الى الاكبر باستخدام الدالة Sort كالتالي:

```
Dim NamesArray(4) As String
```

```
NamesArray(0) = "خالد"  
NamesArray(1) = "نهاد"  
NamesArray(2) = "محمد"  
NamesArray(3) = "كمال"  
NamesArray(4) = "عبد الكريم"
```

```
Array.Sort(NamesArray)
```

```
Me.ListBox1.Items.AddRange(NamesArray)
```

- الدالة AddRange تسمح باضافة جميع عناصر المصفوفة الى اداة ListBox بدلا من استخدام الجمل التكرارية او اضافة كل عنصر على حدا من خلال الدالة Add.

▶ عند تنفيذ الكود السابق نحصل على النتيجة التالية:



كذلك يمكن ترتيب عناصر مصفوفة رقمية كما يبين المثال التالي:

```
Dim NumbersArray() As Integer = {8, 6, 1, 9, 12, 3}
```

```
Array.Sort(NumbersArray)
```

```
For Count As Integer = 0 To NumbersArray.Count - 1  
    Me.lbDetails.Items.Add(NumbersArray(Count))  
Next
```

عند تنفيذ الكود السابق نحصل على عناصر المصفوفة مرتبة بشكل تزايدى (من الاصغر الى الاكبر):





► ويمكننا أيضا قلب عناصر المصفوفة ليصبح اخر عنصر هو اول عنصر وهكذا وذلك

باستخدام الدالة Reverse الخاصة بالمصفوفات كما يبين المثال التالي:

```
Dim myArray() As Integer = {4, 2, 3, 1}
```

'عرض المصفوفة قبل الترتيب'

```
Me.lbDetails.Items.Add("قبل الترتيب")
```

```
For Each Item As Integer In myArray
```

```
    Me.lbDetails.Items.Add(Item)
```

```
Next
```

'عرض المصفوفة بعد الترتيب'

```
Array.Sort(myArray)
```

```
Me.lbDetails.Items.Add("بعد الترتيب")
```

```
For Each Item As Integer In myArray
```

```
    Me.lbDetails.Items.Add(Item)
```

```
Next
```

'عرض المصفوفة بعد القلب'

```
Array.Reverse(myArray)
```

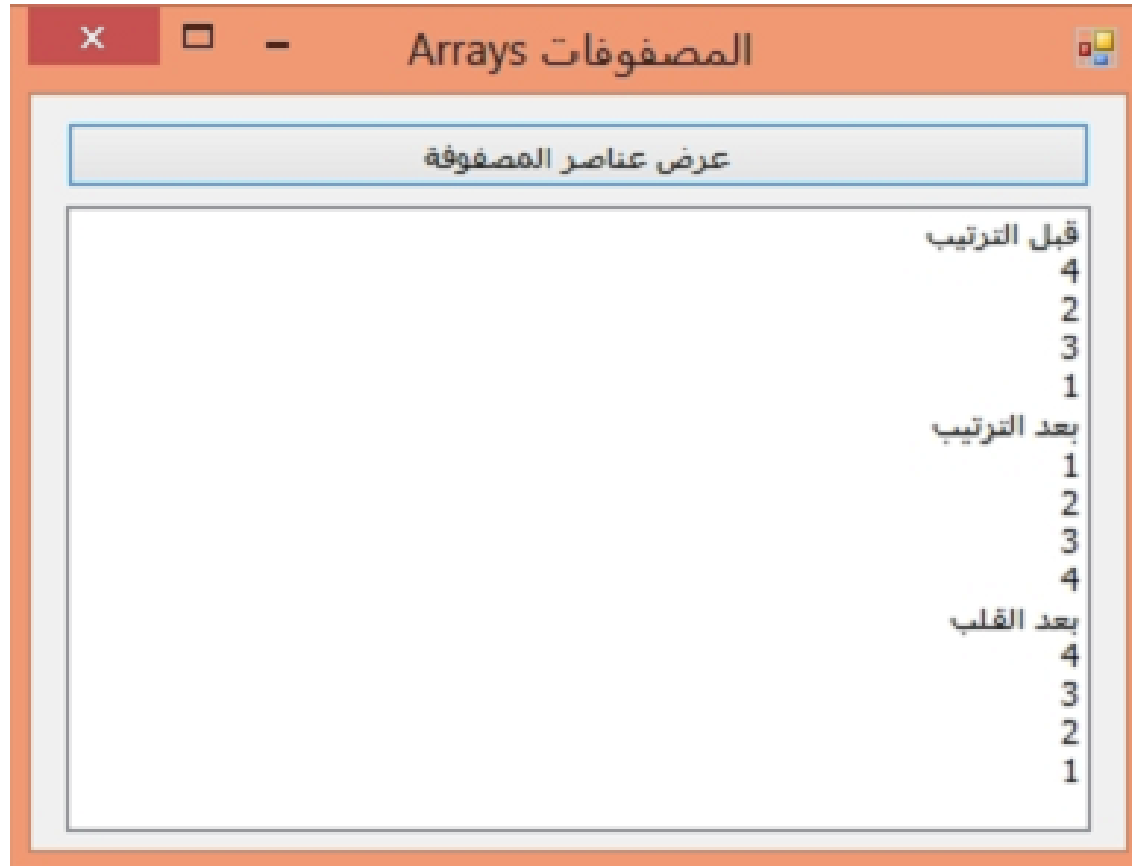
```
Me.lbDetails.Items.Add("بعد القلب")
```

```
For Each Item As Integer In myArray
```

```
    Me.lbDetails.Items.Add(Item)
```

```
Next
```

▶ عند تنفيذ الكود السابق نحصل على النتيجة التالية:



## المصفوفات المتعددة الأبعاد

▶ تمثل المصفوفات المتعددة الأبعاد نوعا آخر من المصفوفات القليلة الاستخدام وأشهرها المصفوفات الثنائية البعد والتي يمكننا تمثيلها على شكل جدول يتكون من صفوف واعمدة

ويتم الاعلان عنها بالصيغة التالية: `Dim TwoDimArray(2, 5) As Integer`

▶ كما يمكننا الاعلان عنها كما يلي:

`Dim TwoDimArray(0 To 2, 0 To 5) As Integer`

▶ المصفوفة الثنائية التي اعلنا عنها اعلاه مكونة من 3 صفوف و 6 اعمدة اي انها تحتوي على 18 عنصرا.

▶ ويمكننا تمثيل عناصر المصفوفة الثنائية على شكل جدول كما يلي:

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)

► يمكننا اسناد القيم لهذه العناصر من خلال رتبته كما يلي:

```
Dim TwoDimArray(2, 5) As Integer
```

```
TwoDimArray(0, 3) = 23
```

```
TwoDimArray(1, 4) = 195
```

```
TwoDimArray(2, 1) = 56
```

الإعلان عن الصفوفة الثنائية'

```
Dim myArray(1, 2) As Integer
```

تعبئة السطر الأول'

```
myArray(0, 0) = 7
```

```
myArray(0, 1) = 2
```

```
myArray(0, 2) = 5
```

تعبئة السطر الثاني'

```
myArray(1, 0) = 3
```

```
myArray(1, 1) = 8
```

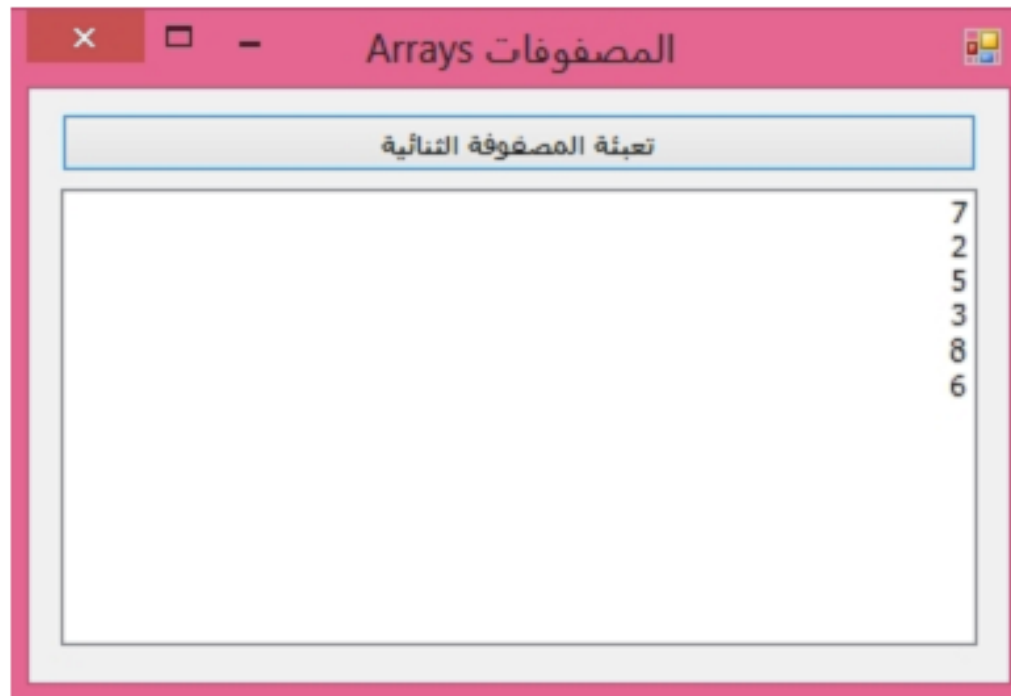
```
myArray(1, 2) = 6
```

الإعلان عن عداد للأسطر وللأعمدة'

```
Dim row, col As Integer
```

```
'التكرار الأول من أجل الأسطر'  
For row = 0 To 1  
    'التكرار الثاني من أجل الأعمدة'  
    For col = 0 To 2  
        lbDetails.Items.Add(myArray(row, col))  
    Next  
Next  
Next
```

▶ عند تنفيذ الكود السابق نحصل على:



Thank you

So Much

تم بحمد الله

