

Chapter Four

Arrays in Visual Basic 6

An array is a collection of simple variables of the same type to which the computer can efficiently assign a list of values. Array variables have the same kinds of names as simple variables. An array is a consecutive group of memory locations that all have the same name and the same type. To refer to a particular location or element in the array, we specify the array name and the array element position number. The individual elements of an array are identified using an index. Arrays have upper and lower bounds and the elements have to lie within those bounds. Each index number in an array is allocated individual memory space and therefore users must evade declaring arrays of larger size than required. We can declare an array of any of the basic data types including variant, user-defined types and object variables. The individual elements of an array are all of the same data type.

Declaring arrays: Arrays may be declared as Public (in a code module), module or local. Module arrays are declared in the general declarations using keyword Dim or Private. Local arrays are declared in a procedure using Dim. Array must be declared explicitly with keyword "As". There are two types of arrays in Visual Basic namely:

Fixed-Size Array: The size of array always remains the same-size doesn't change during the program execution. When an upper bound is specified in the declaration, a Fixed-array is created. The upper limit should always be within the range of long data type.

لا بد هنا أن نعلم بأنه عند كتابة المصفوفة في مكان ما فإن هذا المكان يحدد لنا نطاق استعمال المصفوفة، فمثلاً عند كتابة المصفوفة في داخل إجراء أو حدث معين فإننا نستخدم المصفوفة في داخل ذلك الإجراء فقط، أما إذا قمنا بكتابة المصفوفة في بداية الفورم (أعلى منطقة الكود في الفورم) فإن المصفوفة يمكن استخدامها على طول الكود في الفورم، وفي حالة كتابة المصفوفة بداخل وحدة برمجية فنستطيع استخدام المصفوفة في أي مكان في المشروع. وعند إنشاء أي مصفوفة فأنت باختصار تجيب عن هذه أو توضح المعلومات التالية اسم المصفوفة نوع البيانات ابعاد المصفوفة وعناصر المصفوفة

One Dimension Array: Declaring a fixed-array, if array-Name is the name of an array variable and N is a whole number, then the statement

Dim ArrayName (N) As Var Type

Where, the Dim statement is said to dimension the array and (N) is the range of the array. The array holds either all string values or all numeric values, depending on whether **Var Type** is string or one of the numeric type names.

For example: Dim Num (5) As Integer In the above illustration, num is the name of the array, and the number 6 included in the parentheses is the upper limit of the array. The above declaration creates an array with 6 elements, with index numbers running from 0 to 5.

The numbers inside the parentheses of the individual variables are called **subscripts**, and each individual variable is called a **subscripted variable** or **element**. The elements of an array are assigned successive memory locations. The following figure shows the memory location for the array Num(5)

Num (5)	Num (0)	Num (1)	Num (2)	Num (3)	Num (4)	Num (5)
	1	3	-10	5	3	2

If we want to specify the lower limit, then the parentheses should include both the lower and upper limit along with the To keyword. An example for this is given below.

Dim Num (6) As Integer							
Num (6)	Num (0)	Num (1)	Num (2)	Num (3)	Num (4)	Num (5)	Num(6)
	-	1	3	-10	5	3	2

In the above statement an array of 10 elements (Num(10)) is declared but with indexes running from 1 to 6.

Example: Write a code program to read of one dimensional array A(5). Print the value and position of each element.

A=[2 4 5 6 1]

Solution 1:

```
Dim A(5) as single
```

```
Picture1.cls Picture1.Print "position"; Space (3); "Value of element"
```

```
For I=1 To 5
```

```
A( I )= Val(InputBox( " " ))
```

```
Next I
```

```
For I= 1 to 5
```

```
Picture1.Print I; Space (11); A(I)
```

```
Next
```

Solution 2:

```
Dim A(5) as single
```

```
Picture1.cls Picture1.Print "position"; Space (3); "Value of element"
```

```
For I= 1 To 5
```

```
A( I )= Val(InputBox( " " ))
```

```
Next I For I= 0 to 5
```

```
Picture1.Print I; Space (11); A(I)
```

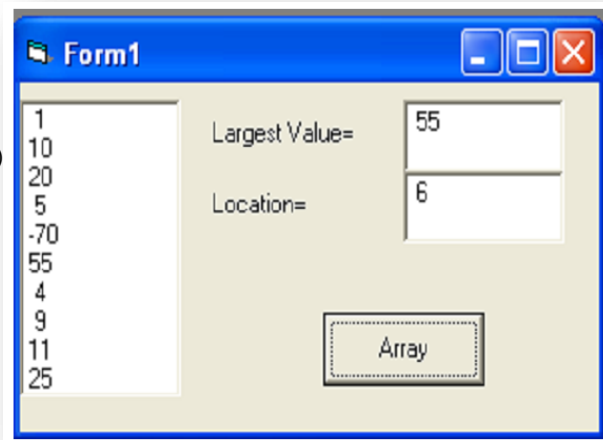
```
Next
```

Note: In solution 2, The type value of array (A) as single, then the default value of A(0)=0, If Type value of array(A) as Variant, then empty value in position A(0)

Example: Suppose A is a one dimension array with 10 elements is entered into list box. Write a program segment to find the location J such that A (J) contains the largest value in A. Display the Largest value and the location into textboxes.

Solution:

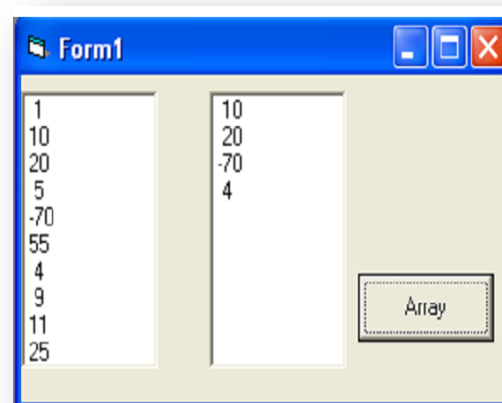
```
Dim A(10) as single
For I=1 To 10
A(I)=Val(list1.list(I-1)) Next Max=A(1)
For J=1 to 10
If A(J)> Max
Then Max=A(J) : P= J
EndIf
Next
Text1.text= Str(Max)
Text2.text= Str(P)
End Sub
```



Example: Suppose A is a one dimension array with 10 elements is entered into listbox. Write a program segment to create the one dimension array (B) contains the even value in array (A). Display the new array (B) into list box2.

Solution:

```
List2.Clear
Dim A(10) As Single, B(10) As Single
For I = 1 To 10
A(I) = Val(List1.List(I - 1))
Next For I = 1 To 10
If A(I) Mod 2 = 0 Then
k = k + 1
B(k) = A(I)
End If
Next
For I = 1 To k
List2.AddItem Str(B(I))
Next
End Sub
```



Example : Suppose X, Y is linear arrays each with 7 elements into inputbox which element from X and Y in one row respectively. Write a program segment to compute the value of (S) from the following formula. Display the value of (S) into textbox.

$$S = \frac{\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2}}{\sqrt{\sum_{i=1}^n X_i y_i}}$$

Solution:

```
Dim X(7) As Single, Y(7) As Single
For I=1 To 7
X(I)= Val (InputBox( "X(i) " ))
Y(I)= Val (InputBox( "Y(i) " ))
Next I
For I=1 To 7
S1=S1+X(I)^2 : S2=S2+Y(I)^2 : S3= S3 +X(I)*Y(I)
```

```

Next I
S= Sqr(S1) * Sqr(S2) / Sqr(S3) Text1.text=Str(S)
End Sub

```

Example: Suppose A is a one dimension array with (10) elements. Write a code program which sorts A so that its elements are increasing and sorters into a new array B. Display the origin array (A) and create array (B) into picturebox which element from A and B in one row respectively.

Solution:

```

Dim A (10 ), B(10)
For I=1 To 10
A ( I ) =Val ( InputBox ( "A=" )) : B ( I ) = A ( I )
Next
For I = 1 To 9
For J = I+1 To 10
If B( J ) < B( I ) Then
D=B( I )
B( I )=B( J )
B( J )=D
EndIf
Next J , I
For I=1 To 10
Picture1.Print A ( I ); space (4); B ( I )
Next
End Sub

```

Two Dimensional Arrays:

Arrays can have multiple dimensions. A common use of multidimensional arrays is to represent tables of values consisting of information arranged in rows and columns. To identify a particular table element, we must specify two indexes: The first (by convention) identifies the element's row and the second (by convention) identifies the element's column.

Tables or arrays that require two indexes to identify a particular element are called two dimensional arrays. The following statement declares a two-dimensional array (3 by 3) within a procedure.

Dim Avg (3, 3) as Single

	Avg (0,0)	Avg (0,1)	Avg (0,2)	Avg (0,3)
Avg (Row, Col.)	Avg (1,0)	Avg (1,1)	Avg (1,2)	Avg (1,3)
	Avg (2,0)	Avg (2,1)	Avg (2,2)	Avg (2,3)
	Avg (3,0)	Avg (3,1)	Avg (3,2)	Avg (3,3)

	2	6	1	0
Avg (3, 3)	3	1	6	-3
	7	3	1	5
	5	4	-2.5	9

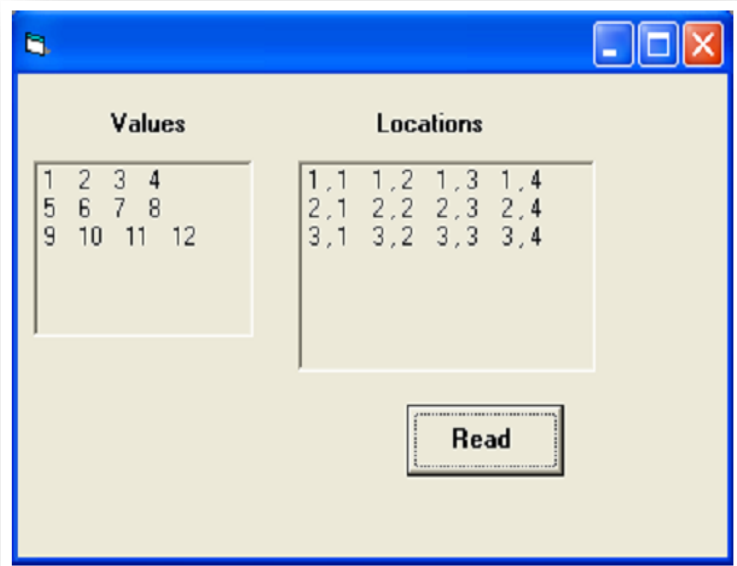
It is also possible to define the lower limits for one or both the dimensions as for fixed size arrays.

المصفوفات متعددة الأبعاد هو تمثيل القيم بشكل جداول التي تتكون من معلومات مرتبة في صفوف وأعمدة. لتحديد عنصر في جدول معين، يجب علينا تحديد مؤشرين: الأول يحدد صف العنصر والثانية يحدد العمود العنصر.

Example: Write a code program to read of two dimensional array A(3,4) on a row by row. Print the value and position of each element.

Solution:

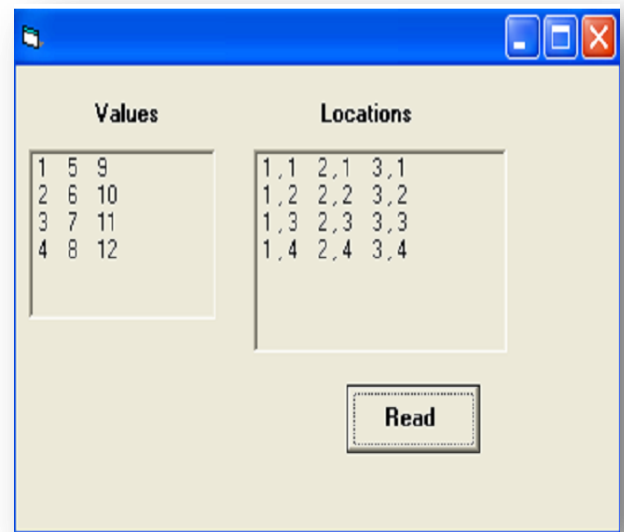
```
Dim A(3,4) As Single
For I=1 To 3           (Rows)
For J= 1 To 4         (Columns)
A(I,J) =Val(InputBox( " " ))
Next J
Next I
For I=1 To 3
For J= 1 To 4
Picture1.Print A(I, J) ; Space(2) ;
Picture2.Print I ; " , " ; J ; Space(2) ;
Next J
Picture1.Print : Picture2.Print
Next I
End sub
```



Example: Write a code program to read of two dimensional array A(3,4) on a column by column. Print the value and position of each element.

Solution: Dim A(3,4) As Single

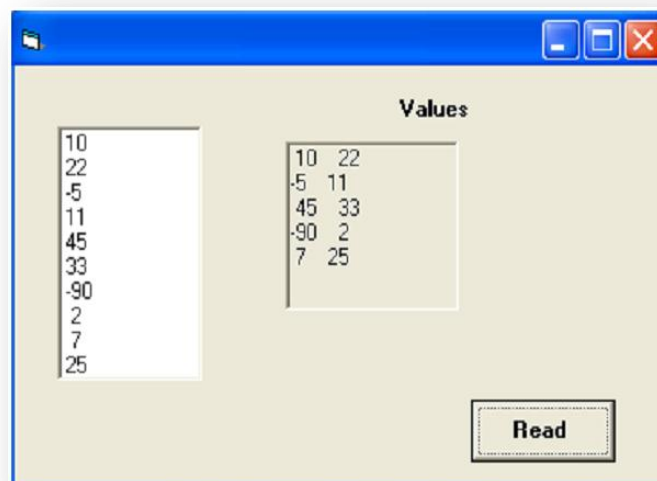
```
For J=1 To 4 (Columns)
For I= 1 To 3 (Rows)
A(I,J) =Val(InputBox( " " ))
Next I
Next J
For J=1 To 4
For I= 1 To 3
Picture1.Print A(I, J) ; Space(2) ;
Picture2.Print I ; " , " ; J ; Space(2) ;
Next I
Picture1.Print : Picture2.Print
Next J
End sub
```



Example: Write a code program to create a two dimensional array N (5X2) into List Box on row by row. Print the values of array N.

Solution:

```
Dim N(5,2) As Single
K=0
For I = 1 To 5
For J=1 To 2
N(I,J)= Val (List1.List (K))
K=K+1
Next J, I
For I=1 To 5
For J= 1 To 2
```

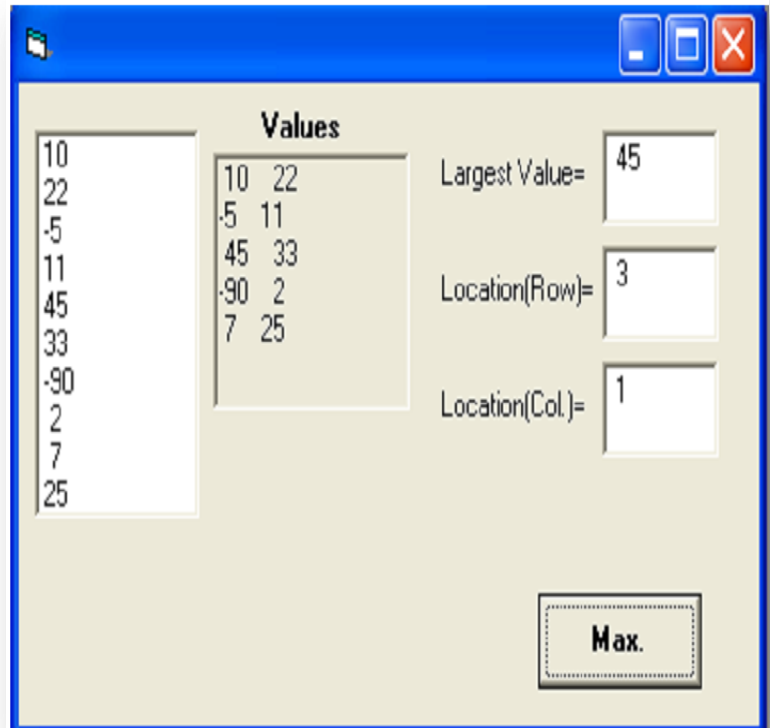


```
Picture1.Print N(I, J) ; Space(2) ;
Next J : Picture1.Print : Next I
```

Example: Suppose N is a (5x2) matrix array is entered into ListBox on row by row. Write a program segment to find the location I and J such that N (I,J) contains the largest value in N. Print the values of array N. Display the Largest value and the location into textboxes.

Solution:

```
Dim N(5,2) As Single
K=0
For I = 1 To 5
For J=1 To 2
N(I,J)= Val (List1.List (K))
K=K+1
Next J, I
Max = N(1, 1): R = 1: C = 1
For I = 1 To 5
For J = 1 To 2
If N(I, J) > Max Then
Max = N(I, J) R = I: C = J
End If
Next J , I
For I = 1 To 5
For J = 1 To 2
Picture1.Print N(I, J); Space(2);
Next J: Picture1.Print: Next I
Text1.Text = Str(Max)
Text2.Text = Str(R)
Text3.Text = Str(C)
```



Example: Write a code program to defined the array H (5,5) Calculate the elements of the numeric array (H). Each element of H is determined by the formula :

$$h_{ij} = i + j - 1$$

Create the one dimensional array X contains the elements of array H(5,5) on row by row. Print the array X into List Box.

Solution:

```
Dim H(5,5) As Single , X(25) As Single
For I=1 To 5
For J=1 To 5
H(I,J)=(I+J-1)
Next J ,I
For I =1 To 5
For J=1 To 5
K=K+1 X(K) =H (I , J)
Next J , I
For I=1 To K
List1.AddItem str(X(I))
Next I
```

Example: Write a code program to read the elements of the array T(5,3) on a row by row. Calculate the SUM of elements in each row and stored in column 4. Print a new array T(5,4) and the sum of all individual row sums, the cumulative sum for all rows

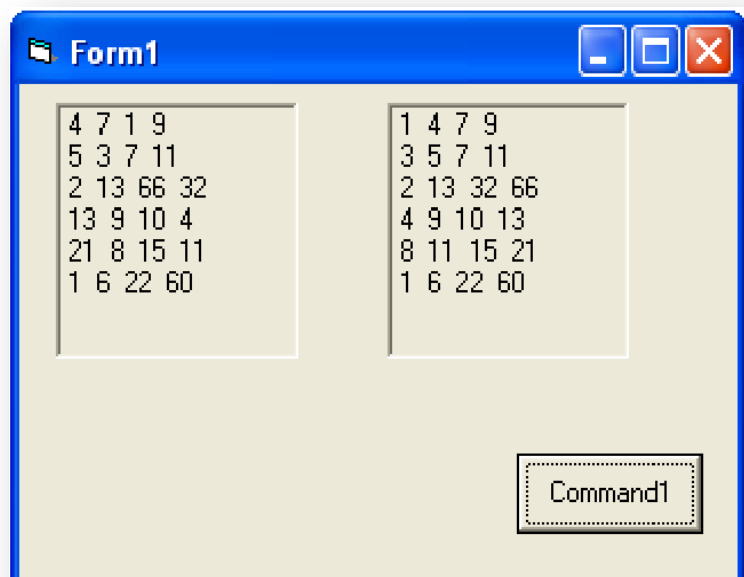
Solution:

```
Dim T (5, 4) As Single
For I =1 To 5
For J=1 To 3
T(I,J)=Val (InputBox( “ ” ))
Next J, I
For I=1 To 5
Sum=0
For J=1 To 3
Sum=Sum + T(I ,J)
Next J
T (I,4) = Sum
Total=Total +T(I,4)
Next I
For I=1 To 5
For J=1 To 3
Picture1.print T (I , J);
Next J: Picture1.Print : Next I
Text1.text=Str(Total)
End Sub
```

Example: : Suppose W is a two dimension array with (6,4). Write a code program which sorts W on row by row so that its elements are increasing (Ascending) and sorters into a same array. Display the new array (W) into picturebox which element.

Solution:

```
Dim W(6,4) as Single
For I=1 To 6
For J=1 To 4
W ( I, J)= Val (InputBox( “ ” ))
Next J , I
For I =1 To 6
For J= 1 To 3
For K= J+1 To 4
If W(I,K) < W(I,J) Then
C=W(I,J)
W(I,J)=W(I,K) : W(I,K)=C
End If
Next K , J, I
For I=1 To 6
For J =1 To 4
Picture1.print W(I,J);
Next J : Picture1.Print : Next I
End Sub
```



Dynamic Array:

The size of the array can be changed at the run time- size changes during the program execution. In actual practice, the amount of data that a program will be processing is not known in advance. Programs should be flexible and incorporate a method for handling varying amounts of data. Visual basic makes this possible with the statement

ReDim ArrayName (N) As Var Type

This can use variables or expression when indicating the subscript range. However, ReDim statements can only be used inside procedures.

المصفوفات المتغيرة أو المصفوفات التفاعلية. هذا النوع من المصفوفات لا يحتاج لتحديد عدد عناصر المصفوفة في مرحلة الكود وإنما في مرحلة تشغيل البرنامج ولتعريف مصفوفة تفاعلية نقوم بتحديد أسم ونوع المصفوفة في مرحلة الكود وبدون ذكر عدد العناصر للمصفوفة، فمثلاً إذا كان لدينا مصفوفة لدرجات الحرارة باسم Temperatures نقوم بتعريفها كالتالي:

Dim Temperatures() As Single

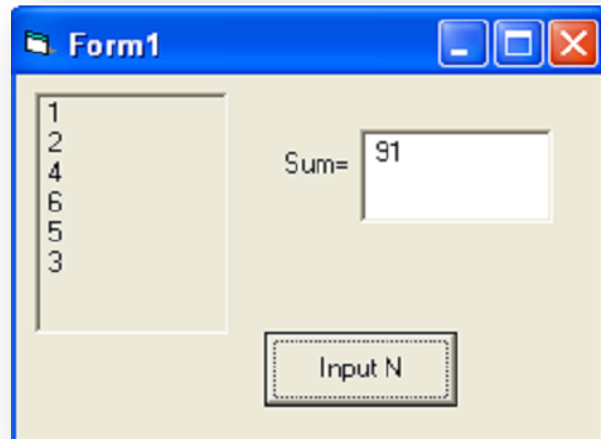
بدون ذكر عدد عناصرها في مرحلة الكود. بعد كود تعريف المصفوفة نقوم بكتابة كود آخر يسمح للمستخدم بكتابة عدد عناصر المصفوفة في مرحلة تشغيل البرنامج Runtime، بطريقة أو بأخرى يمكننا السماح للمستخدم بكتابة عدد عناصر المصفوفة باستخدام الكود بدون كتابة عدد العناصر في مرحلة الكود وإنما إسناد العدد إلى متغير معين.

Example: Suppose a one dimension array A with N elements is entered into inputbox. Write a visual basic program segment which is calculated the sum of the squares:

$$\text{Sum} = \sum_{i=1}^N (A_i)^2$$

Solution:

```
Dim N
N= Val (InputBox( "N=" ))
ReDim A(N)
For I=1 To N
A(I) =Val (InputBox( " " ))
Sum=Sum+A(I)^2
Next
Text1.Text=Str (Sum)
```



Example: An MxN matrix array A is entered into input box. Write a visual basic program segment which is calculated the SUM of elements in each Column and Stored in a one dimension Z. Print the arrays A and Z and the sum of all elements of array Z.

Solution:

```
Dim M, N
M=Val (Text1.Text) : N= Val (Text2.Text)
ReDim A (M, N), Z(N)
For I=1 To M
For J=1 To N
A(I,J) = Val (InputBox( " " ))
Next J , I
For J=1 To N
```

```

Sum=0
For I=1 To M
Sum=Sum+ A(I, J)
Next I
Z(J)=Sum
Total=Total +Z(J)
Next J
For I =1 To M
For J=1 To N
Picture1.Print A(I, J);
Next J: picture1.print: Next I
For I=1 To N
Picture2.print Z(I)
Next
Text3.text=Str(Total)

```

Example: Suppose a string array Name(N) and a numeric array Grade(N,8). Write A code program to read the student name and student grade for 8 subjects. Extend the program to calculate student grades average, stored the average grades into (column number 9). Sort the average grades so that its elements are increasing. Display the student name and the average grades for all students.

Solution:

```

Dim N As Single
N= Val (InputBox( "Number of Student" ))
ReDim Name (N) As String,Grade(N,9) As Single
For I=1 To N
Name (I)=Val (InputBox( "Name" ))
For J=1 To 8
Grade (I, J) =Val (InputBox( "grade" & Str(J) ))
Next J, I
For I=1 To N
Sum=0
For J=1 To 8
Sum=Sum+ Grade (I, J)
Next J
Grade (I,9)=Sum / 8
Next I
For I=1 To N-1
For J=I+1 To N
If Grade (J, 9) < Grade(I, 9) Then
For K=1 To 9
C=Grade (I,K)
Grade (I,K)=Grade (J,K)
Grade (J,K)=C
D=Name (I)
Name (I)= Name (J)
Name (J)=D
Next K
End If
Next J, I
For I=1 To N
Picture1.Print Name (I)

```

```

For J= 1 To 9
Picture2.Print Grade(I,J);
Next J: Picture2.print
Next I

```

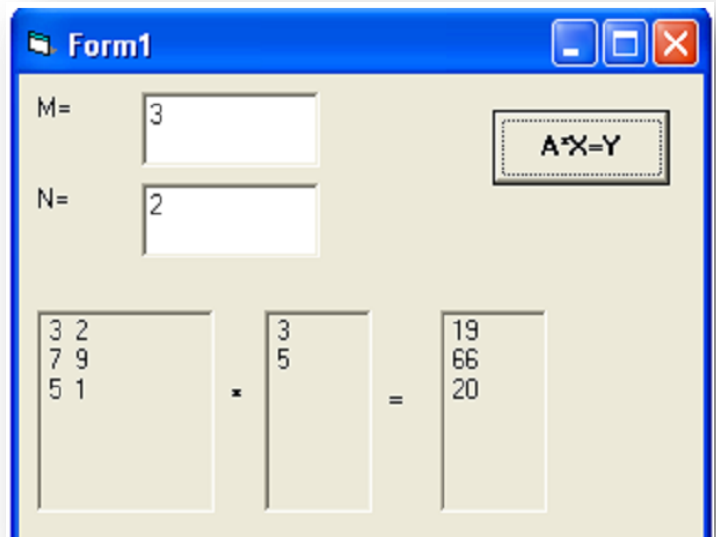
Example: Suppose a numeric array A having M row and N columns, and a numeric array X having N elements. Write a code Program to calculate a numeric array Y by carrying out the following operations. Print a new array

Solution:

```

Dim M, N
M=Val (Text1.Text) : N= Val (Text2.Text)
ReDim A (M, N), X(N), Y(M)
For I=1 To M
For J=1 To N
A(I,J) = Val (InputBox( "" ))
Picture1.Print A(I, J);
Next J:
Picture1.Print:
Next I
For I=1 To N
X(I)=Val (Inputbox( "" ))
Picture2.Print X(I)
Next I
For I=1 To M
For J=1 To N
Y(I)=Y(I)+A(I, J)*X(J)
Next J, I
For I=1 To M
Picture3.Print Y(I)
Next I

```



Note: Each element of Y is determined from the following formula:

$$Y(1) = A(1,1)*X(1)+A(1,2)*X(2)$$

$$Y(2) = A(2,1)*X(1)+ A(2,2)*X(2)$$

$$Y(3) = A(3,1)*X(1)+A(3,2)*X(2)$$

Example 6-18: Suppose a numeric array (A) having (5 row and 3 columns), and that (B) is a numeric array having (3 rows and 5 columns). Write a code Program to calculate the elements of the numeric array C, where a numeric array C is determined from the following formula: $[C] = [A] * [B]$ Print a numeric array C.

Solution: Each element of array C is determined by $[C]_{m,n} = [A]_{m,k} * [B]_{k,n}$

```

Dim A(5,3) , B(3,4) , C(5,4)
For I=1 To 5
For J=1 To 3
A(I,J)=Val (InputBox( "A" ))
Next J , I
For I=1 To 3
For J=1 To 4
B(I,J)=Val (InputBox( "B" ))
Next J , I

```

```

For I=1 To 5
For J=1 To 4
For K=1 To 3
C(I,J)=C(I,J) + A( I,K ) * B( K,J )
Next K, J, I
For I=1 To 5
For J=1 To 4
Picture1.Print C(I,J);
Next J: Picture1.Print: Next I

```

Note: Each element of C is determined from the following formula:

$$C(1,1) = A(1,1)*B(1,1) + A(1,2)*B(2,1) + A(1,3)*B(3,1)$$

$$C(2,2) = A(2,1)*B(1,2) + A(2,2)*B(2,2) + A(2,3)*B(3,2)$$

خلاصة الفصل:

✓ إنشاء مصفوفة نقوم بتعريف المصفوفة باستخدام الكلمة Dim كالتالي:

Dim A(9) As String ✓

✓ إنشاء مصفوفة عامة نعرفها بداخل وحدة برمجية Module بكتابة الكلمة Public كالتالي:

Public (9) As String ✓

✓ لتعامل مع عناصر المصفوفة نستخدم الحلقات التكرارية للتعامل مع عناصر المصفوفة (9) For I=1 to

✓ المصفوفة التفاعلية هي المصفوفة التي نستطيع تغيير عدد عناصر وقت تنفيذ البرنامج ويمكننا تعريف هذه المصفوفة بتعريف مصفوفة بالشكل المعروف مع تحديد نوع عناصرها لكن مع عدم كتابة عدد العناصر، وإذا كانت المصفوفة ببعدين أو بثلاثة أبعاد فنكتب الفاصلة بداخل القوس بين كل بعد والآخر بدون كتابة عدد العناصر. ثم نقوم بإعادة تعريف المصفوفة في مرحلة الكود وإسناد عدد العناصر إلى متغير يقوم المستخدم بتعبئته في مرحلة تشغيل البرنامج كالتالي: لنفرض بأن المتغير هو Cars

ReDim Temperatures(Cars)

✓