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
```
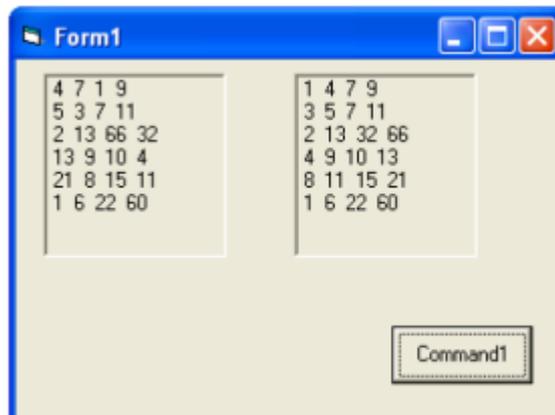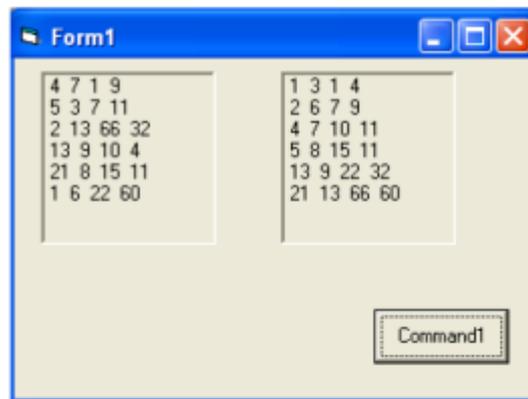
```
For I=1 To 6
For J =1 To 4
Picture1.print W(I,J);
Next J : Picture1.Print : Next I
```

Example   Suppose W is a two dimension array with (6,4). Write a code program which sorts W on column by column so that its elements are increasing (Ascending) and stores 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 J =1 To 4
For  I= 1 To 5
For K= I+1 To 6
If W(K,J) < W(I,J) Then
C=W(I,J)
W(I,J)=W(K,J) :  W(K,J)=C
End If
Next K , I, J
For I=1 To 6
For J =1 To 4
Picture1.print W(I,J);
Next J : Picture1.Print : Next I
```

Form1

```
4 7 1 9          1 3 1 4
5 3 7 11         2 6 7 9
2 13 66 32       4 7 10 11
13 9 10 4        5 8 15 11
21 8 15 11       13 9 22 32
1 6 22 60        21 13 66 60
```

Command1

- **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.

Example   Suppose a one dimension array A with N elements is entered into inputbox. Write visual basic program segment which is calculated the sum of the squares: $Sum=\sum_{i-1}^{N}(A_i)^2$

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)
```

Form1
```
1
2
4          Sum=   91
6
5
3
```
Input N
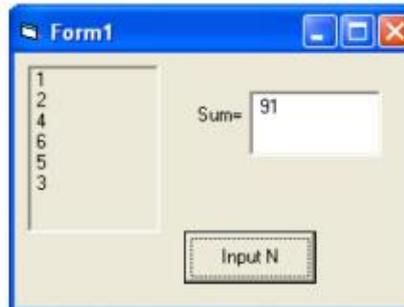
Example   An MxN matrix array A is entered into input box. Write a visual basic program segment which is calculated the SUM of elements in each Column and Stored in a one dimension Z. Print the arrays A and Z and the sum of all elements of array Z.

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)
```
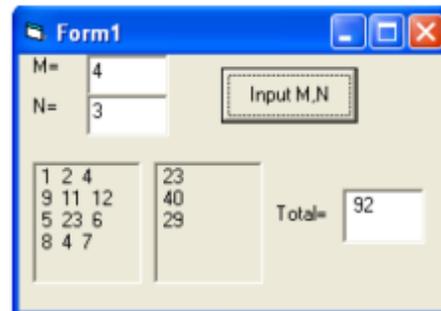
Form1
```
M=   4
N=   3                    Input M,N

1 2 4      23
9 11 12    40      Total=  92
5 23 6     29
8 4 7
```

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
```