04/11/2013

Informatica per la Finanza
Informatica per le Scienze Economiche
Informatica e Programmazione per il Turismo

Vettori e matrici

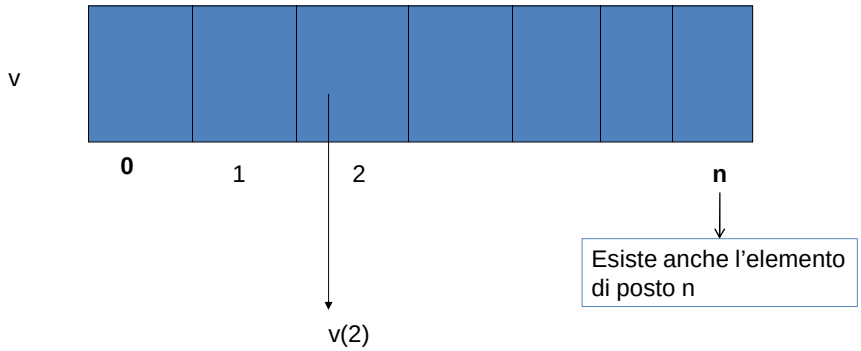
- Sono gruppi di variabili dello stesso tipo nelle quali si fa riferimento agli elementi mediante un indice:
  - possono avere una (vettori) o più dimensioni (matrici)
    - Spesso si userà matrice per entrambi i casi
  - la dimensione può essere fissa o dinamica
  - gli indici partono da 0:
    - Si può fare in modo che gli indici partano sempre da 1, come le celle di Excel, scrivendo all'inizio del modulo fuori da tutte le routine:

```
Option Base 1
```

    - si può modificare il punto di partenza e di arrivo

Una matrice monodimensionale

- Dim v(n)




```
Option Explicit
Sub prv()
  Dim v(7) As Integer, i As Integer
  For i = 0 To 7
     v(i) = InputBox("dammi intero " _
                     & i & ": ")
     v(i) = v(i) * 2
     MsgBox ("v(" & i & ") = " & v(i))
  Next
End Sub
```

Carica un vettore di 8 elementi, raddoppia i valori in esso contenuti, visualizza i valori attraverso una Message Box

Esempio

- Caricare la matrice V con i primi 5 valori numerici della colonna A
  - la colonna contiene almeno 5 valori

```
Option Explicit
Sub prv()
   Dim v(4) As Double, i As Integer
   Dim rg As Integer 'riga

   rg = 1
   Do While i < 5
      If IsNumeric(Cells(rg, 1)) Then
        v(i) = Cells(rg, 1)
        i = i + 1
      End If
      rg = rg + 1
   Loop

End Sub
```

il vettore ha posizioni da 0 a 4 incluso!

Esempio

- Caricare la matrice V con i primi 5 valori numerici della colonna A
  - la colonna contiene almeno 5 valori

```
Option Explicit
Option Base 1
Sub prv()
   Dim v(5) As Double, i As Integer
   Dim rg As Integer 'riga

   rg = 1
   i = 1 'il vettore non ha la pos. 0
   Do While i <= 5
      If IsNumeric(Cells(rg, 1)) Then
        v(i) = Cells(rg, 1)
        i = i + 1
      End If
      rg = rg + 1
   Loop

End Sub
```

il vettore ha posizioni da 1 a 5 incluso!

Matrici

- Il singolo elemento di una matrice è una variabile del tipo con cui è formata la matrice
  - Nelle matrici di tipo **`Variant`** possono essere presenti sia elementi di tipo numerico che di tipo stringa
- Le matrici son passate solo per riferimento (**`ByRef`**) ad una routine
- Le matrici NON POSSONO essere
  - Assegnate in blocco
  - Confrontate in blocco

Matrici statiche

- La dichiarazione avviene specificando il numero di elementi di ogni dimensione:

```
Dim nomeMatrice(dim1, ...,dimN) As Tipo
```

  - Se si cerca di accedere ad un elemento oltre le dimensioni viene generato un errore di esecuzione
  - Le variabili di tipo numerico sono inizializzate a **`0`**, quelle di tipo **`Variant`** ad **`Empty`**
- Ci si riferisce ad un elemento attraverso le sue coordinate:

```
nomeMatrice(coord1, ...,coordN)
```

```
Option Explicit
Sub provaVettore()
  Dim vet(3) As Double, matr(4, 2) As Double
  Dim i, j As Integer

  For i = 0 To 3
    vet(i) = Rnd
  Next
  For i = 0 To 4
    For j = 0 To 2
      matr(i, j) = Rnd
    Next
  Next

  Range("A1").Value = "vettore"
  For i = 0 To 3
    Cells(1, i + 2).Value = vet(i)
  Next
  Range("A3").Value = "matrice"
  For i = 0 To 4
    For j = 0 To 2
      Cells(3 + j, i + 2).Value = matr(i, j)
    Next
  Next
End Sub
```

	0	1	2
0	0,00	0,00	0,00
1	0,00	0,00	0,00
2	0,00	0,00	0,00
3	0,00	0,00	0,00
4	0,00	0,00	0,00

La funzione **Rnd** genera numeri casuali fra 0 ed 1escluso

Esempio

- caricare nella matrice M i valori nell'intervallo A1..E5

```
Option Explicit
Option Base 1
Sub prv()
   Dim M(5, 5) As Double
   Dim i As Integer, j As Integer

   For i = 1 To 5
     For j = 1 To 5
        M(i, j) = Cells(i, j)
     Next
   Next

End Sub
```

Esempio

- Scrivere un sub in VBA che
  - carica un matrice 2x3 di double di nome Mt
  - somma tutti gli elementi contenuti nella matrice
    - la prima coordinata indica le righe, la seconda le colonne

```
        0       1       2
Mt  0 (0,0)   (0,1)   (0,2)
        2       5     6.5
    1 (1,0)   (1,1)   (1,2)
      -6.5      8       3
```

```
Option Explicit
Sub matrix()
  Dim Mt(1, 2) As Double
  Dim som As Double
  Dim i As Integer
  Dim j As Integer
  som = 0
  For i = 0 To 1
   For j = 0 To 2
     Mt(i, j) = _
     InputBox("valore (" & _
       i & "," & j & "): ")
     som = som + Mt(i, j)
   Next
  Next
  Cells(1, 1) = som
End Sub
```


Matrici statiche

- In VB è possibile indicare esplicitamente l'intervallo di variazione degli indici:

  Dim *nmMatr* (*da1* To *a1*, ..., *daN* To *aN*)

- E' possibile conoscere quali sono i valori degli indici minimo e massimo di una matrice grazie a

  LBound(*NomeMatrice*, *dimensione*)

  UBound(*NomeMatrice*, *dimensione*)

  - Se non specificato ***dimensione*** vale 1
  - E' meglio usare i valori di queste funzioni per gestire le matrici

Matrici statiche

```
Sub provaIndici()
Dim Vet(5 To 9) As Double
Dim i As Integer

For i = LBound(Vet) To UBound(Vet)
  Vet(i) = i * 10
Next

For i = LBound(Vet) To UBound(Vet)
    Cells(6, i + 2).Value = Vet(i)
Next
End Sub
```


Esempio

```
Option Explicit
Sub matrix()
  Dim Mt(3 To 5, 2 To 3) As Double         '3x2
  Dim i As Integer, j As Integer

  For i = LBound(Mt) To UBound(Mt)
   For j = LBound(Mt, 2) To UBound(Mt, 2)
     Mt(i, j) = InputBox("valore (" & i & "," & j & "): ")
   Next
  Next

End Sub
```

Matrici statiche

```
Option Explicit
Option Base 1
Sub matrix()
  Dim Fl(9) As Variant
  Dim i As Integer
  Dim rg As Integer, cl As Integer

  i = 1
  For rg = 1 To 3
    For cl = 1 To 3
       Fl(i) = Cells(rg, cl)
       i = i + 1
    Next
 Next

End Sub
```

```
Option Explicit
Option Base 1
Sub prv()
  Dim M(3, 3) As Double, i As Integer,
   j As Integer

  For i = LBound(M, 1) To UBound(M, 1)
   For j = LBound(M, 2) To UBound(M, 2)
       M(i, j) = Cells(i, j)
    Next
  Next

End Sub
```

Scrivere la sub che legge dal foglio di lavoro i valori dalla cella A1 alla C3 e li carica
- nel vettore FL (9)
- Matrice M(3,3)

Matrici statiche

```
Option Explicit
Sub prvIndiciMat()
 Dim matr(4, 3) As Double
 Dim i, j As Integer
  For i = LBound(matr) To UBound(matr)
    For j = LBound(matr, 2) To UBound(matr, 2)
      matr(i, j) = Rnd
    Next
  Next
  Range("A9").Value = "matrice"
  For i = LBound(matr, 1) To UBound(matr)
    For j = LBound(matr, 2) To UBound(matr, 2)
      Cells(10 + j, i + 2).Value = matr(i, j)
    Next
  Next
End Sub
```

Carica una matrice con numeri casuali (con **Rnd**)
E ne scrive il contenuto a partire dalla cella B10

Esempio

```
Option Explicit
Option Base 1
Sub orla()
  Dim m(3, 2) As Double, n(4, 3) As Double
  Dim i As Integer, j As Integer, som As Double

  For i = LBound(m, 1) To UBound(m, 1)
    For j = LBound(m, 2) To UBound(m, 2)
       m(i, j) = Cells(i, j)
    Next
 Next

  For i = LBound(m, 1) To UBound(m, 1)
    som = 0
    For j = LBound(m, 2) To UBound(m, 2)
       n(i, j) = m(i, j)
       som = som + m(i, j)
    Next
    n(i, j) = som
 Next

 For i = LBound(n, 2) To UBound(n, 2)
    som = 0
    For j = LBound(n, 1) To UBound(n, 1) - 1
       som = som + n(j, i)
    Next
    n(j, i) = som
 Next
End Sub
```

2	3	5
4	5	9
6	7	13
12	15	27

m		
n		

Carica la matrice m 3x2 con i valori contenuti nel foglio di lavoro e quindi crea una matrice n 4x3 che contiene:
- la matrice m
- nella riga 4 la somma dei valori delle colonne
- nella colonna 3 la somma dei valori delle righe

Matrici statiche

Scrivere una funzione che legge il contenuto delle celle da A1 ad A5 e lo memorizza in un vettore lo raddoppia e lo scrive nelle celle da B7 in poi

```
Option Explicit
Option Base 1
Sub matri()
  Dim celle(5) As Double, i As Integer

 For i = LBound(celle) To UBound(celle)
     celle(i) = Cells(i, 1).Value
 Next
 For i = LBound(celle) To UBound(celle)
   Cells(i + 6, 2).Value = celle(i) * 2
  Next
End Sub
```

Matrici statiche

- Scrivere la sub che legge il contenuto del foglio di lavoro dalla cella A1 alla cella C4 e carica una matrice 4x3 con i valori numerici interi positivi contenuti nelle celle. Se il valore della cella non è intero e positivo scrive nella matrice -1

Matrici statiche

```
Option Explicit
Option Base 1
Function eIntero(v As Double) As Boolean
   Dim val As Integer
   val = v
   eIntero = (val = v)
End Function

Sub prv()
  Dim Mat(4, 3) As Integer, i As Integer, j As Integer

  For i = LBound(Mat, 1) To UBound(Mat, 1)
    For j = LBound(Mat, 2) To UBound(Mat, 2)
      If IsNumeric(Cells(i, j).Value) Then
        If eIntero(Cells(i, j).Value) And Cells(i, j).Value > 0 Then
          Mat(i, j) = Cells(i, j)
        Else
          Mat(i, j) = -1
        End If
      Else
         Mat(i, j) = -1
      End If
    Next
  Next
End Sub
```

Esempio (sol. alternativa)

```
Option Explicit
Option Base 1
Sub matrix()
  Dim Mat(4, 3) As Integer
  Dim rg As Integer, cl As Integer

  For rg = 1 To 4
    For cl = 1 To 3
       If IsNumeric(Cells(rg, cl)) Then
          If (Cells(rg, cl) > 0) Then
            If CInt(Cells(rg, cl)) = Cells(rg, cl) Then
              Mat(rg, cl) = Cells(rg, cl)
            Else
              Mat(rg, cl) = -1
            End If
          Else
            Mat(rg, cl) = -1
          End If
       Else
         Mat(rg, cl) = -1
       End If
    Next
  Next
End Sub
```


Matrici dinamiche

- Le matrici dinamiche possono cambiare la loro dimensione durante l'esecuzione

  Dim *nmMatrDin*() As *Tipo*

  – Prima dell'uso la matrice va dimensionata:

  ReDim *nmMatrDim*(*dim1*, ..., *dimN*)

  - Una matrice può essere ridimensionata più volte durante l'esecuzione
    - Ogni ridimensionamento fa perdere il contenuto precedente a meno di usare l'opzione **Preserve**.
  - **Dim*i*** può essere solo l'estremo superiore o un intervallo **(valoreIniziale To ValoreFinale)**

Matrici dinamiche

```
Sub matrDin()
   Dim nD() As Double
   Dim i, j As Integer
   ReDim nD(8)
   For i = LBound(nD) To UBound(nD)
      nD(i) = i
      Cells(1, i + 1).Value = nD(i)
   Next
   ReDim nD(4, 8 To 13)
   For i = LBound(nD, 1) To UBound(nD, 1)
     For j = LBound(nD, 2) To UBound(nD, 2)
       nD(i, j) = i * j + 100
       Cells(3 + i, j + 1) = nD(i, j)
     Next
   Next
End Sub
```

Modifica estremo superiore prima dimensione

Modifica estremo Inferiore e superiore della seconda dimensione


```
Option Explicit
Option Base 1
Sub allunga()
   Dim vt() As Double, i As Integer
   Dim vDim As Integer, j As Integer
   ReDim vt(4)
   For i = LBound(vt) To UBound(vt)
     vt(i) = Rnd: Cells(1, i) = vt(i)
   Next
   vDim = UBound(vt): ReDim vt(8, 2)
   For i = LBound(vt) To UBound(vt)
   For j = LBound(vt, 2) To UBound(vt, 2)
     vt(i, j) = Rnd
   Next: Next
   For i = LBound(vt) To UBound(vt)
     For j = LBound(vt, 2) To UBound(vt, 2)
     Cells(2 + i, j) = vt(i, j)
   Next: Next
End Sub
```

Matrici Dinamiche

- L'uso di **`ReDim`** con l'opzione **`Preserve`** impedisce la modifica:
  - Delle dimensioni della matrice eccetto l'ultima
    - In un vettore a due dimensioni posso modificare solo la seconda
    - **In un vettore ad una dimensione posso modificare la dimensione**
  - Del numero delle dimensioni
  - Del tipo della matrice

Matrici Dinamiche

```
Sub matrDinPre()
   Dim nD() As Double
   Dim i, j As Integer

   ReDim nD(8)
   For i = LBound(nD) To UBound(nD)
      nD(i) = i
      Cells(1, i + 1).Value = nD(i)
   Next

   ReDim Preserve nD(12)
   For i = LBound(nD) To UBound(nD)
       nD(i) = i * 1000
       Cells(10, i + 1) = nD(i)
   Next
End Sub
```

Matrici Dinamiche

```
Option Explicit
Option Base 1
Sub dinamico()
  Dim Mt() As Double, i As Integer
  Dim j As Integer

  ReDim Mt(4, 5 To 9)

  For i = LBound(Mt, 1) To UBound(Mt, 1)
    For j = LBound(Mt, 2) To UBound(Mt, 2)
       Mt(i, j) = Rnd
    Next
  Next

  ReDim Preserve Mt(4, 5 To 6)
  For i = LBound(Mt, 1) To UBound(Mt, 1)
    For j = LBound(Mt, 2) To UBound(Mt, 2)
       Cells(i, j) = Mt(i, j)
    Next
  Next
End Sub
```

Non è possibile modificare il valore
Del limite inferiore della seconda
dimensione

Questa dimensione non si può comunque modificare

Matrici: passaggio parametri

- Le matrici devono essere passate per riferimento ad una routine

```
Sub carica(v() As Double)
  Dim i As Integer
  For i = LBound(v) To UBound(v)
    v(i) = InputBox("dammi un valore")
  Next
End Sub
Sub stampaFoglio(v() As Double, cx As Integer, cy As Integer)
  Dim i As Integer
  For i = LBound(v) To UBound(v)
    Cells(cx, cy + i) = v(i)
  Next
End Sub
Sub x()
  Dim vt(5) As Double, vq() As Double
  Call carica(vt): Call stampaFoglio(vt, 1, 1)
End Sub
```

Matrici: passaggio parametri

```
Option Explicit
Option Base 1

Sub carica(v() As Integer)
 Dim i As Integer
 For i = LBound(v)_
             To UBound(v)
  v(i) = InputBox("intero: ")
 Next
End Sub
```

```
Sub stampa(v() As Integer)
  Dim i As Integer
  Dim s As String
  s = ""
  For i = LBound(v) _
             To UBound(v)
     s = s & v(i) & " "
  Next
  Cells(10, 5) = s
  MsgBox (s)
End Sub

Sub princ()
   Dim vt(4) As Integer
   Call carica(vt)
   stampa vt
End Sub
```


Matrici: passaggio parametri

- Scrivere la funzione che ha come parametro una matrice 2x2 e restituisce il contenuto della matrice raddoppiato
- E la somma di tutti gli elementi in essa contenuti

Matrici: passaggio parametri

```
Option Explicit
Option Base 1
Function raddoppia(ByRef mt() As Integer)
  Dim i As Integer, j As Integer

  For i = LBound(mt, 1) To UBound(mt, 1)
    For j = LBound(mt, 2) To UBound(mt, 2)
       raddoppia = raddoppia + mt(i, j)
       mt(i, j) = mt(i, j) * 2
    Next
  Next
End Function

Sub prv()
  Dim M(2, 2) As Integer, somma As Integer
  Dim i As Integer, j As Integer

  'inizializzo la matrice
  For i = LBound(M, 1) To UBound(M, 1)
    For j = LBound(M, 2) To UBound(M, 2)
       M(i, j) = Cells(i, j)
    Next
  Next

  somma = raddoppia(M)
End Sub
```