



Le Matrici - Statiche e Dinamiche

Una **Matrice** (o Array) è una collezione di variabili che condividono lo stesso nome e tipo di dati e costituiscono un comodo metodo per memorizzare un certo numero di dati nello stesso contenitore. Diversamente da quanto accade con i tipi di dati definiti dall'utente, gli elementi di una matrice devono essere *tutti dello stesso tipo*. Per esempio se si crea una matrice di tipo Integer, tutti gli elementi al suo interno devono essere di tipo Integer, in altre parole le matrici vengono utilizzate per rappresentare elenchi o tabelle in cui tutti i dati siano dello stesso tipo. Una matrice permette di memorizzare ed elaborare una serie di dati utilizzando una sola variabile che permette oltre alla riduzione complessiva di nomi di variabili da gestire, la possibilità di utilizzare dei cicli per semplificare l'elaborazione dei diversi elementi che la compongono.

Combinando matrici e cicli (tipicamente cicli *For Next* e *ForEach*) è possibile scrivere procedure di poche istruzioni che elaborano grandi quantità di dati, che se si dovesse usare nomi distinti per le variabili, le stesse procedure potrebbero richiedere centinaia di istruzioni. La forma più semplice di una matrice non è altro che un elenco di elementi che vengono definite semplici o unidimensionali

■ Dichiarazioni di Matrici

Per dichiarare una matrice è necessario prevedere il limite superiore (l'ultimo e più grande numero di indice), mentre il limite inferiore (il primo e il più piccolo numero di indice) è facoltativo e se si omette questo argomento, viene determinato dalle impostazioni del modulo, che di default è uguale a 0. Per riuscire a tenere traccia delle dimensioni di un array siano essi statici o dinamici il VBA prevede due funzioni, *Lbound* e *Ubound*, che restituiscono il valore minimo e massimo per gli indici di un array. La sintassi generica per queste funzioni è

Lbound(NomeArray [, dimensione])

Ubound(NomeArray [, dimensione])

L'argomento "*dimensione*" è un numero intero che specifica per quale dimensione dell'array si vuole ottenere il limite minimo o massimo. Se non viene specificato VBA restituisce l'estremo relativo alla prima dimensione dell'array. La funzione "*Ubound*" restituisce il valore della posizione più alta per la dimensione indicata di una matrice, mentre "*Lbound*" è l'opposto e restituisce il valore più basso possibile, mentre il valore di ritorno per entrambe queste funzioni è un dato di tipo Integer. Quando un array utilizza un solo indice viene chiamato "*unidimensionale*", mentre una matrice "*multidimensionale*" utilizza più di un indice.

Va ricordato che la dimensione di una matrice è il numero totale di elementi che è determinato dal prodotto delle sue dimensioni e per determinare la sua dimensione massima si deve tenere presente che gli indici partono da 0 e il valore massimo è definito dal numero indicato nella dichiarazione. È possibile dichiarare un array multidimensionale di tipo Byte con la sintassi: `Dim K As Byte` e ottenere la dimensione della matrice utilizzando la funzione "*Ubound*", oppure utilizzare la funzione *Lbound* per ottenere il valore più basso possibile per l'indice. Se l'array ha un solo elemento, il valore di ritorno di *Ubound* è uguale a 0 e *Lbound* restituisce sempre 0 finché l'array non è stato inizializzato, anche se non ha elementi. Esempio per un array monodimensionale: `Dim MyArray1 (5) As String`

Limite Inferiore = `Lbound (MyArray1)` 'Restituisce il valore 0

Limite Superiore = `Ubound (MyArray1)` 'Restituisce il valore 5

Dimensioni = `Ubound (MyArray1) - Lbound (MyArray1) + 1` 'Restituisce il valore 6

Si deve tenere presente che quando si utilizzano queste funzioni per un array multi-dimensionale, si deve specificare la dimensione in base al quale il limite inferiore o superiore devono essere determinati.



Dovrebbe risultare familiare l'uso dell'istruzione **Dim** che abbiamo visto usare per le variabili, ma viene usata anche per la dichiarazione di matrici, in effetti la parola chiave Dim è un'abbreviazione di *dimension* e nella versione moderna del VBA la parola chiave *Dim* permette di dichiarare matrici sia unidimensionali che a più dimensioni. La sintassi generica per la dichiarazione è la seguente

Dim Nome_variabile ([Indici]) [As Tipo]

Nome_variabile rappresenta un qualsiasi nome per la matrice che risponda alle regole di VBA per i nomi delle variabili e l'argomento *Indici* rappresenta le dimensioni della matrice, che è possibile dichiarare con un numero di dimensioni fino a 60. Per una matrice unidimensionale si usa una sola dichiarazione di indici, per una a due dimensioni se ne includono due separate da una virgola e via così fino a raggiungere il numero di dimensioni desiderato per la matrice. La sezione Indici ha la seguente sintassi

[minimo To] massimo [, [minimo To] massimo]

Minimo specifica il valore minimo valido per l'indice della matrice e *massimo* quello più alto, si deve tenere presente che è richiesto soltanto il limite superiore, in quanto l'indicazione del valore minimo per l'indice è opzionale. Se si specifica solo il limite superiore, VBA numera gli elementi della matrice coerentemente con l'impostazione *Option Base*, se è stato specificato *Option Base 1*, VBA numera gli elementi della matrice da 1 al valore massimo, in caso contrario gli elementi della matrice vengono numerati da 0 al valore massimo. Per esempio

Dim gennaio (1 to 31) As String

Dim gennaio(31) As String

Nella seconda riga del listato poco sopra si presume *Option Base 1* e specificando la parte *minimo To* si facilita la comprensibilità del codice e si semplifica la ricerca di errori, rendendo i programmi più affidabili. Dichiarare la parte *minimo To* permette anche di specificare come valore di partenza per l'indice un valore diverso da 0 o 1. Per esempio potreste voler creare una matrice i cui elementi siano numerati da 5 a 10 o da -5 a 0, a seconda della particolare operazione che intendete compiere. Come nelle normali dichiarazioni di variabili, è possibile dichiarare per una matrice un particolare tipo di dati specificando la parte *As tipo*, in questo modo ogni elemento della matrice sarà del tipo specificato, che può essere qualsiasi tipo di dato VBA valido

Finora abbiamo visto matrici dotate di indici che iniziano da 0 (*zero-based*), che secondo questa convenzione di numerazione, l'indice del primo elemento di qualsiasi dimensione di una matrice è 0, una matrice di 5 elementi ha dunque indici che vanno da 0 a 4. Evidentemente la numerazione *zero-based* può generare confusione perché l'elemento di indice zero in realtà indica il primo componente della matrice l'indice 4 indica il quinto elemento e così via. Sarebbe molto più comodo se gli elementi di una matrice fossero numerati a partire da 1 invece che da 0. In questo caso l'indice 1 indicherebbe il primo elemento della matrice, l'indice 5 il quinto e così via. Fortunatamente VBA permette di specificare il numero iniziale per gli elementi di una matrice, infatti è possibile specificare se si desidera che gli indici della matrice partano da 0 o da 1 con l'istruzione **Option Base** che presenta questa sintassi

Option Base 0|1

L'istruzione *Option Base* permette di impostare 0 o 1 come indice di base della matrice, in mancanza di questa istruzione VBA assegna per default la numerazione degli elementi della matrice a partire da 0. L'istruzione *Option Base* deve essere inserita nell'area delle dichiarazioni di un modulo prima di qualsiasi dichiarazione di variabile, costante o procedura e non può comparire all'interno di una procedura. Ecco due esempi



Option Base 0 'Impostazione predefinita

Option Base 1 'Gli indici della matrice partono da 1

■ Matrici Unidimensionali

Un array monodimensionale è un elenco di elementi che hanno una singola riga o una singola colonna, ad esempio, le vendite trimestrali di una società durante l'anno (la dimensione sarà la riga o la colonna dei 4 trimestri dell'anno per il quale i dati di vendita saranno inseriti). Di seguito vediamo alcune dichiarazioni per creare un array:

Per creare un array con 7 elementi, con i numeri di indice 0-6 (il valore predefinito del limite inferiore è 0) si usa la sintassi: *Dim prova1(6) As String* oppure *Dim prova1(0 To 6) As String*. In questo secondo caso il limite inferiore deve essere specificato in modo esplicito, con la parola chiave *To*. Supponiamo di avere un array che ha 20 elementi in cui i numeri di indice vanno da 1 a 20 si può usare la seguente dichiarazione: *Dim prova2(1 To 20) As Integer*.

Nel caso si abbia un array con 51 elementi in cui i numeri di indice vanno da 100 a 150 si può usare la seguente dichiarazione: *Dim prova2(100 To 150) As Integer*.

In sostanza i formati per una dichiarazione di matrice unidimensionale, con e senza specificare il limite inferiore possono essere rispettivamente:

Dim ArrayName(Index) as DataType, oppure

Dim ArrayName(First_Index To Last_Index) as DataType

Pertanto possiamo riassumere che il nome è legato al fatto che un elenco di dati è assimilabile a una linea tracciata su un foglio, che ha una sola dimensione, e una sola lunghezza, ed è quindi *unidimensionale* o monodimensionale.

10	Elemento 0
10,5	Elemento 1
11	Elemento 2
11,6	Elemento 3
11,9	Elemento 4

Fig. 1

La **Figura 1** rappresenta una matrice *unidimensionale* e ogni dato memorizzato nella matrice è detto elemento della matrice e come si vede in figura la matrice contiene 5 elementi, ciascuno dei quali contiene un numero di tipo *Double*, noterete anche che gli elementi della matrice sono numerati da 0 a 4 per un totale di 5 e che la numerazione parte da zero. Per accedere al dato memorizzato in un certo elemento si usa il nome della matrice seguito dal numero (detto indice) dell'elemento desiderato racchiuso tra parentesi tonde. Per esempio se il nome della matrice fosse *NumMat* la seguente istruzione assegnerebbe il valore 11,6 alla variabile *alex*

alex = NumMat(3)

Nell'esempio sopra riportato 3 è l'*indice della matrice*, è racchiuso tra parentesi tonde e non è separato da spazi dal nome della matrice. Visto che la numerazione degli elementi parte da zero, l'elemento a cui fa riferimento l'istruzione è in realtà il numero 4 di *NumMat* e quando si esegue questa istruzione VBA legge il valore 11,6 dall'elemento della matrice indicato e lo memorizza nella variabile *alex* come per qualsiasi assegnamento.



L'indice viene anche utilizzato tutte le volte che si vuole salvare un valore in un certo elemento della matrice, l'istruzione seguente, per esempio, memorizza nel secondo elemento della matrice il valore 12

NumMat(2) = 12

Quando VBA esegue l'istruzione sopra riportata, scrive il valore 12 nell'elemento della matrice indicato, sostituendone il contenuto precedente, esattamente come per qualsiasi assegnazione, potete usare gli elementi di una matrice all'interno di una espressione VBA come se si trattasse di una qualsiasi variabile. Vediamo degli esempi di codice per un array monodimensionale:

```
Sub demo1()  
Dim K1(3) As String  
K1(0) = "Primo "  
K1(1) = "Secondo "  
K1(2) = "Terzo "  
K1(3) = "Quarto "  
MsgBox K1(0) & "- " & K1(1) & "- " & K1(2) & "- " & K1(3)  
End Sub
```

E riporta un messaggio come il seguente:

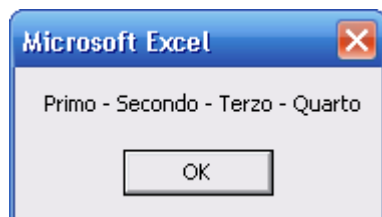


Fig. 2

```
Sub demo2()  
Dim K2(-3 To 2) As String  
K2(-3) = "Uno"  
K2(-2) = "Due"  
K2(-1) = "Tre"  
K2(0) = "Quattro"  
K2(1) = "Cinque"  
K2(2) = "Sei"  
MsgBox K2(-3) & " - " & K2(-2) & " - " & K2(-1) & " - " & K2(0) & " - " & K2(1) & " - " &  
K2(2)  
End Sub
```

E riporta un messaggio come il seguente:

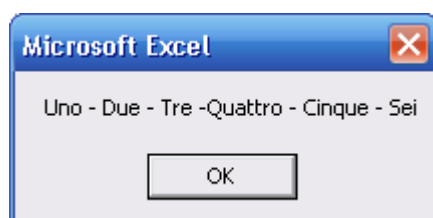


Fig. 3



Esempio: gestire gli array monodimensionali con un ciclo

	A	B
1	Nome	Peso
2	Mario	85 Kg.
3	Piero	96 Kg.
4	Alice	47 Kg.
5	Rosa	45 Kg.

Fig. 4

```
Sub demo3()  
Dim i As Integer  
Dim nome(2 To 5) As String, peso(2 To 5) As Single  
For i = 2 To 5  
    nome(i) = InputBox("Inserisci il Nome")  
    peso(i) = InputBox("Inserisci il Peso")  
    Sheet4.Cells(i, 1) = nome(i)  
    Sheet4.Cells(i, 2) = peso(i) & " Kg."  
Next i  
End Sub
```

■ Matrici Multidimensionali

Le matrici unidimensionali vanno bene finché si tratta di rappresentare semplici elenchi di dati, spesso però nei vostri programmi vi troverete a dover rappresentare tabelle di informazioni in cui i dati sono organizzati in righe e colonne, più o meno come accade in un foglio di Excel, per farlo dovete ricorrere a una matrice multidimensionale, che rappresentiamo in **figura 5**

	Colonna 0	Colonna 1	
	10	12,4	Elemento 1,0
Elemento 0,1	10,5	15,2	
	11	18	
Elemento 0,3	11,6	13	Elemento 1,3
	11,9	19,8	

Fig. 5

Una matrice bidimensionale ha 2 dimensioni, comprendenti righe e colonne e utilizza due indici, uno rappresenta le righe e l'altro rappresenta le colonne e vengono utilizzati quando è necessario specificare un elemento con due attributi, ad esempio, le vendite trimestrali di una società nel corso degli ultimi 5 anni (una dimensione saranno i 4 trimestri dell'anno e la seconda dimensione saranno i 5 anni). Un array bidimensionale appare con una forma a tabella, con più righe e colonne e la sintassi è:

Dim ArrayName (Index1, Index2) As DataType oppure
Dim ArrayName (First_Index1 To Last_Index1, First_Index2 To Last_Index2) As DataType.

Per esempio se si dichiara un array bidimensionale si utilizza la seguente forma:

Dim prova (7,9) As Long oppure *Dim prova (7,1 To 9) As Long*



Tenendo presente che il limite inferiore può essere specificato in modo esplicito in una o entrambe le dimensioni.

Semplificando quanto esposto possiamo affermare che le matrici *multidimensionali* devono il loro nome al fatto di avere più di una dimensione, la lunghezza (il numero di righe) e la larghezza (il numero di colonne), come si vede in in *figura 5*, la matrice ha due colonne (numerate 0 e 1) e 5 righe numerate da 0 a 4 per un totale di 10 elementi. Per accedere agli elementi di matrici multidimensionali si usa un indice, cioè si usano i riferimenti di riga e colonna per identificare un certo elemento. L'indicizzazione di una matrice bidimensionale assomiglia molto al metodo per identificare le celle di un foglio di lavoro di Excel, la prima dimensione della matrice corrisponde alle colonne del foglio di lavoro e la seconda alle righe. Se chiamiamo *NewMat* la matrice di *figura 5* la seguente istruzione assegna alla variabile *alex* il valore 12,4 (prima riga, seconda colonna della matrice)

alex = *NewMat*(1,0)

Analogamente, la seguente istruzione memorizza il valore 5,5 nella seconda riga della prima colonna della matrice

NewMat(0,1) = 5,5

Da notare che in entrambe le precedenti istruzioni gli indici della matrice sono racchiuse tra parentesi e che le coordinate della colonna e della riga sono separate da virgole. Le matrici possono avere più di due dimensioni, la figura sottostante (**Fig. 6**) mostra una matrice a tre dimensioni con una lunghezza, una larghezza e una profondità (per modo di dire)

0		1		2	
Colonna 0	Colonna 1	Colonna 0	Colonna 1	Colonna 0	Colonna 1
Elemento 0,0,0	Elemento 0,1,0	Elemento 1,0,0	Elemento 1,1,0	Elemento 2,0,0	Elemento 2,1,0
Elemento 0,0,1	Elemento 0,1,1	Elemento 1,0,1	Elemento 1,1,1	Elemento 2,0,1	Elemento 2,1,1
Elemento 0,0,2	Elemento 0,1,2	Elemento 1,0,2	Elemento 1,1,2	Elemento 2,0,2	Elemento 2,1,2
Elemento 0,0,3	Elemento 0,1,3	Elemento 1,0,3	Elemento 1,1,3	Elemento 2,0,3	Elemento 2,1,3
Elemento 0,0,4	Elemento 0,1,4	Elemento 1,0,4	Elemento 1,1,4	Elemento 2,0,4	Elemento 2,1,4

Fig. 6

Potete pensare ad una matrice tridimensionale come all'insieme delle pagine di un libro in cui ogni pagina contenga una tabella con lo stesso numero di righe e colonne. Nella figura sopra riportata (*Figura 6*) la nostra matrice è di tre pagine (per riprendere l'esempio del libro) numerate da 0 a 2 e ogni pagina contiene una tabella di 2 colonne e 5 righe, e nelle caselle di ogni elemento sono riportate le coordinate nella matrice.



Esempio di una matrice a due dimensioni

	A	B	C	D	E	F
1		Anno	Anno1	Anno2	Anno3	Anno4
2	prova1	500	410	320	250	150
3	prova2	520	440	330	280	180
4	prova3	545	425	335	275	175
5	prova4	595	485	300	205	105

Fig. 7

```
Sub demo2()  
Dim i As Integer, n As Integer  
Dim vendi(1 To 4, 1 To 5) As Long  
vendi(1, 1) = 500  
vendi(2, 1) = 520  
vendi(3, 1) = 545  
vendi(4, 1) = 595  
vendi(1, 2) = 410  
vendi(2, 2) = 440  
vendi(3, 2) = 425  
vendi(4, 2) = 485  
vendi(1, 3) = 320  
vendi(2, 3) = 330  
vendi(3, 3) = 335  
vendi(4, 3) = 300  
vendi(1, 4) = 250  
vendi(2, 4) = 280  
vendi(3, 4) = 275  
vendi(4, 4) = 205  
vendi(1, 5) = 150  
vendi(2, 5) = 180  
vendi(3, 5) = 175  
vendi(4, 5) = 105  
For i = 1 To 4  
For n = 1 To 5  
Foglio1.Cells(i + 1, n + 1) = vendi(i, n)  
Next n  
Next i  
End Sub
```

■ Matrici Statiche e Dinamiche

In VBA possono essere create due tipi di matrici, matrici a dimensione fissa o statiche e matrici dinamiche. Una matrice che ha un numero fisso di elementi è una matrice di dimensioni fisse e viene utilizzato quando si conosce il numero preciso di elementi che verranno contenuti nella matrice, però la maggior parte delle volte sarà necessario creare array dinamico, perché non si sa l'esatta dimensione della matrice richiesta all'inizio e serve una certa flessibilità per modificare il numero di elementi della matrice.



Normalmente la dichiarazione di una matrice indica a VBA l'estensione delle varie dimensioni che provvede ad allocare una quantità di memoria sufficiente per tutti gli elementi della matrice, nel caso della matrice di *figura 1* VBA alloccherebbe memoria per 5 interi mentre per la matrice di *figura 6* VBA alloccherebbe memoria per 10 elementi. VBA riserva spazio in memoria per tutti gli elementi della matrice finché la relativa variabile esiste, matrici di questo tipo vengono dette **Statiche** perché il numero di elementi non cambia. Scegliere la dimensione di una matrice può essere complicato se non sapete a priori quanti dati dovrà ospitare o se la quantità di dati raccolti è molto variabile.

Se a volte gli elementi da memorizzare sono 100 e altre volte 10, potenzialmente vi troverete a sprecare lo spazio per memorizzare 90 elementi inutili (è la differenza tra il numero maggiore di elementi e il minore), per casi come questi il VBA prevede un tipo particolare di matrice, detto **dinamico**. Le matrici dinamiche sono definite in questo modo perché è possibile modificarne il numero di elementi durante l'esecuzione del programma e le matrici dinamiche se associate a una corretta programmazione possono crescere o stringersi per far posto esattamente al numero di elementi necessari eliminando lo spreco di spazio. Per modificare le dimensioni di una matrice dinamica si usa l'istruzione **Redim**.

E' possibile dichiarare una variabile dinamica con le dimensioni dell'indice vuoto e successivamente dimensionare o ridimensionare la matrice dinamica che è già stata dichiarata, utilizzando l'istruzione *ReDim*. Per ridimensionare un array, è necessario fornire il limite superiore, mentre il limite inferiore è facoltativo e se non si menziona verrà determinato dalle impostazioni del modulo, che di default è *Option Base 0*. Per esempio è possibile dichiarare la matrice *prova1* come una matrice dinamica in questo modo: *Dim prova1 () As String*, per ridimensionare le dimensioni della matrice e portarla a 3 elementi (specificare *Option Base 1*), si utilizza l'istruzione *Redim* in questo modo: *ReDim prova1 (3) As String*, si può utilizzare una matrice dinamica invece di una a dimensione fissa, se si desidera regolare il numero di record nel database in fase di esecuzione.

```
Option Base 1
Sub prova2()

Dim prova1() As String
ReDim prova1(3) As String
prova1(1) = "Lunedì"
prova1(2) = "Martedì"
prova1(3) = "Mercoledì"
MsgBox prova1(1) & " - " & prova1(2) & " - " & prova1(3)
End Sub
```

Altro esempio di array dinamico

```
Option Base 1
Sub prova3()

Dim prova2() As String
ReDim prova2(3) As String
prova2(1) = "Lunedì"
prova2(2) = "Martedì"
prova2(3) = "Mercoledì"
ReDim prova2(4) As String
prova2(4) = "Giovedì"
MsgBox prova2(1) & " - " & prova2(2) & " - " & prova2(3) & " - " & prova2(4)
End Sub
```




■ Array dinamici – usare la parola chiave Preserve con l'istruzione ReDim

Quando un array viene ridimensionato utilizzando l'istruzione ReDim, i suoi valori si potrebbero perdere, per ovviare a questo possibile inconveniente e garantire che i valori della matrice non siano persi, si utilizza la parola chiave "*Preserve*" l'istruzione *ReDim*, che consentirà di conservare i dati esistenti nella matrice. Ad esempio, prima viene usata l'istruzione ReDim per dimensionare la matrice prova2: "*ReDim prova2 (3) As String*" che veniva popolata con 3 elementi, ora per ridimensionare la matrice e consentire la memorizzazione di 4 variabili senza perdere i dati esistenti, è necessario utilizzare l'istruzione "*ReDim Preserve myArray (4) As String*". Vedere l'esempio sotto riportato

```
Option Base 1
Sub demo4()
Dim New_mat() As String
ReDim New_mat(3) As String

New_mat(1) = "Primo"
New_mat(2) = "Secondo"
New_mat(3) = "Terzo"

ReDim Preserve New_mat(4) As String
New_mat(4) = "Quarto"
MsgBox New_mat(1) & " - " & New_mat(2) & " - " & New_mat(3) & " - " & New_mat(4)
End Sub
```

Si deve tenere presente che la dichiarazione Redim presenta le seguenti caratteristiche:

- Non può cambiare il tipo di dati della matrice
- Non può modificare il numero di dimensioni in un array
- Se si utilizza la parola chiave "*Preserve*", è possibile ridimensionare solo l'ultima dimensione della matrice, in modo che in un array multidimensionale gli stessi limiti devono essere specificati per tutte le altre dimensioni.

Vedi esempio sotto riportato. Si suppone di avere una tabella come mostrato nella figura sotto riportata che riporta le vendite per anno suddivise per reparto. Con le istruzioni che abbiamo appena visto si andrà a ridimensionare la matrice prima estendendola a 4 anni per poi ridurla a 2 anni.

	A	B	C	D	E	F
1		Anno	Anno1	Anno2	Anno3	Anno4
2	Reparto 1	500	410	320	250	150
3	Reparto 2	520	425	350	240	180
4	Reparto 3	545	485	390	290	190
5	Reparto 4	595	413	300	260	135

Fig. 8



```
Sub demo_6()  
Dim annO As Integer, reP As Integer, ampliaM As Integer, limitI As Integer, limitS As Integer,  
limitSR As Integer, rng As String  
  
Dim matric() As Integer  
'la 1° dimensione specifica i 4 trimestri, la seconda dimensione specifica 2 anni di dati  
'per i quali saranno inseriti i dati di vendita  
ReDim matric(1 To 4, 1 To 2)  
'Stabilire i limiti superiori delle 2 dimensioni  
limitI = UBound(matric, 1)  
limitS = UBound(matric, 2)  
  
For annO = 1 To limitS  
For reP = 1 To limitI  
matric(reP, annO) = InputBox("Inserisci le vendite del reparto " & reP & " per l'anno " & annO)  
Foglio1.Cells(reP + 1, annO + 1) = matric(reP, annO)  
Next reP  
Next annO  
  
If MsgBox("Continuare al prossimo anno?", vbQuestion + vbYesNo, "Confirmation") = vbYes Then  
'con ReDim aumentiamo i dati di vendita a tre anni, dai 2 precedenti, per i 4 trimestri  
ReDim matric(1 To 4, 1 To limitS + 1)  
'Determinare il limite superiore della seconda dimensione, dopo il ridimensionamento  
limitSR = UBound(matric, 2)  
'si cicla per ciascuno dei 4 trimestri, inserendo i dati di vendita per l'anno aggiuntivo  
For ampliaM = 1 To limitI  
  
matric(ampliaM, limitSR) = InputBox("Inserire i dati di vendita " & ampliaM & " per l'anno " &  
limitSR)  
Foglio1.Cells(1, 4) = "Anno 3"  
Foglio1.Cells(ampliaM + 1, limitSR + 1) = matric(ampliaM, limitSR)  
Next ampliaM  
  
End If  
If MsgBox("Continuare con la riduzione della matrice?", vbQuestion + vbYesNo, "Confermare") =  
vbYes Then  
'è possibile inserire qualsiasi condizione, come cancellare i dati immessi in precedenza  
With Sheets("Foglio1")  
rng = .Name & "!" & .Cells(2, "B").Address & ":" & .Cells(limitI + 1, limitS + 2).Address  
End With  
Foglio1.Range(rng).ClearContents  
'ora la matrice conterrà i dati di vendita per due anni  
ReDim matric(1 To 2, 1 To 2)  
For annO = 1 To 2  
For reP = 1 To 2  
matric(reP, annO) = InputBox("Inserisci i dati di vendita " & reP & " per l'anno " & annO)  
Foglio1.Cells(reP + 1, annO + 1) = matric(reP, annO)  
Next reP  
Next annO  
End If  
End Sub
```



Fig. 9

Terminata la fase di inserimento dei 2 anni, viene richiesta la conferma per continuare ai prossimi anni



Fig. 10

Inseriti i dati dell'anno richiesto ritorna un messaggio per ridurre la matrice



	A	B	C	D	E	F	G
1		Anno	Anno1	Anno 3	Anno3	Anno4	
2	Reparto 1	320	410	240	250	150	
3	Reparto 2	550	420	350	240	180	
4	Reparto 3	650	480	360	290	190	
5	Reparto 4	230	420	350	260	135	



Fig. 11

Cliccando sul pulsante Si verranno cancellati i dati, ridotta la matrice e richiesti nuovi dati

	A	B	C	D	E	F	G
1		Anno	Anno1	Anno 3	Anno3	Anno4	
2	Reparto 1				250	150	
3	Reparto 2				240	180	
4	Reparto 3				290	190	
5	Reparto 4				260	135	

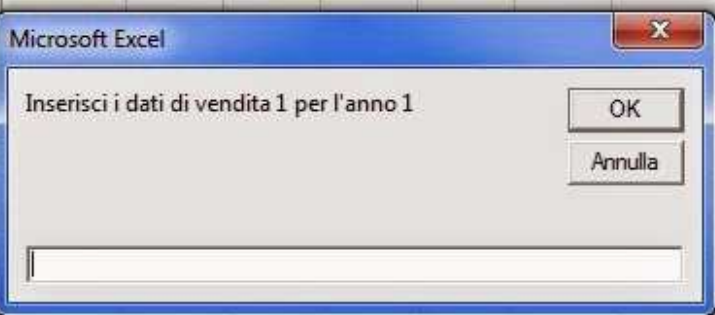


Fig. 12

Una volta inseriti i dati richiesti la matrice presenta questo aspetto

	A	B	C	D
1		Anno	Anno1	Anno 3
2	Reparto 1	320	360	
3	Reparto 2	350	35	
4	Reparto 3			
5	Reparto 4			

Fig. 13

