

## Array pluridimensionali

Finora abbiamo considerato *array monodimensionali*, i quali richiedono un solo indice per l'individuazione di un elemento.

Il Fortran mette a disposizione anche *array pluridimensionali*, in cui l'organizzazione degli elementi è di tipo matriciale. In questo caso, sono necessari più indici per identificare un elemento nell'array.

Questo tipo strutturato permette di affrontare tutte quelle situazioni in cui è necessario lavorare con matrici, tabelle, ecc.

## Array bidimensionali

Per definire un array bidimensionale, è necessario specificare:

- il nome della variabile array
- il tipo cui gli elementi appartengono
- il numero degli elementi presenti nelle due dimensioni (cardinalità di riga e cardinalità di colonna dell'array)

**! un array di 10x10 elementi reali**

**INTEGER :: numrighe,numcolonne**

**REAL :: mat**

**PARAMETER (numrighe=10,numcolonne=10)**

**DIMENSION mat(numrighe,numcolonne)**

Che differenza c'è rispetto ad un array monodimensionale di 100 elementi di tipo real ?

Per accedere ai singoli elementi di un array bidimensionale, è necessario specificare il nome della variabile array e gli indici di riga e di colonna che individuano l'elemento desiderato.

	1	2	3	...	8	9	10	
1	1.0	3.6	-1.5	...	3.8	0.0	2.7	
2	0.1	7.2	4.8	...	3.0	6.4	9.3	

  

				...				
10	7.5	7.7	4.7	...	6.5	9.3	7.1	

## Inizializzazione di un array bidimensionale

```
PROGRAM InizArray2D
  IMPLICIT NONE

  INTEGER :: mat,i,j,numrighe,numcolonne
  PARAMETER(numrighe=10,numcolonne=10)
  DIMENSION mat(righe,colonne)

  ! Inizializzazione dell'array mat
  DO i=1,numrighe
    DO j=1,numcolonne
      mat(i,j)=0
    END DO
  END DO

END PROGRAM
```

## Cardinalità e riempimento

Anche nel caso di array bidimensionali, è possibile definire, per ogni dimensione, una variabile intera che costituirà, a seconda del caso, il numero di righe o di colonne effettivamente presenti nell'array (*riempimento di riga e riempimento di colonna*).

```
PROGRAM LeggiArray2D
  IMPLICIT NONE
  INTEGER :: mat,numrighe,numcolonne
  INTEGER :: i,j,riempirighe,riempcolonne
  PARAMETER(numrighe=10,numcolonne=10)
  DIMENSION mat(numrighe,numcolonne)

  ! Lettura dell'array mat
  WRITE(*,*) "Quante righe ?"
  READ(*,*) riempirighe
  WRITE(*,*) "Quante colonne ?"
  READ(*,*) riempcolonne

  DO i=1,riempirighe
    DO j=1,riempcolonne
      WRITE(*,*)"Valore mat(",i,",",j,"): "
      READ(*,*) mat(i,j)
    END DO
  END DO

END PROGRAM
```

## Esempio: prodotto matrice x vettore

Si debba calcolare  $A \bullet v$ , dove  $A$  è una matrice reale di  $M$  righe ed  $N$  colonne e  $v$  è un array di  $N$  elementi reali. Sia  $w$  l'array che debba contenere il risultato.

```
PROGRAM ProdMatVet
  IMPLICIT NONE
  INTEGER :: numrig,numcol,M,N
  REAL :: A,v,w
  PARAMETER(numrig=50,numcol=50)
  DIMENSION A(numrig,numcol),v(numcol),w(numcol)

  ! Lettura di A
  ! Lettura di v

  DO i=1,M
    w(i)=0.0
    DO j=1,N
      w(i)=w(i)+A(i,j)*v(j)
    END DO
  END DO

END PROGRAM
```