Le Matrici (Array Bidimensionali)

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Vettori bidimensionali (Matrici)

- Il concetto di vettore come collezione di elementi consecutivi, può essere esteso immaginando che gli elementi siano a loro volta dei vettori: si ottiene così un vettore bidimensionale o matrice.
- La definizione di matrice ricalca pienamente quella del vettore:

tipo nome [dim1] [dim2];

tipo può essere un qualunque tipo semplice,

- *dim1, dim2*; racchiusi tra parentesi quadre, definiscono il numero di elementi di ogni dimensione.

Vettori bidimensionali (Matrici)

Esempio:

matrice bidimensionale di numeri interi formata da tre righe e 4 colonne:

```
int a[3][4];
```

	Colonna 0	Colonna 1	Colonna 2	Colonna 3
Riga 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Riga 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Riga 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

indice di colonna

nome del vettore

indice di riga

RAM

a	0FFC	A[0][0]
	1000	A[0][1]
	1004	A[0][2]
	1008	A[0][3]
	100C	A[1][0]
	1010	A[1][1]
	1014	A[1][2]
	1018	A[1][3]
	101C	A[2][0]
	1030	A[2][1]
	1034	A[2][2]
	1038	A[2][3]

offset

A[2][2]=Indirizzo RAM=IndBase+[(indRig*NumCol)+indCol]

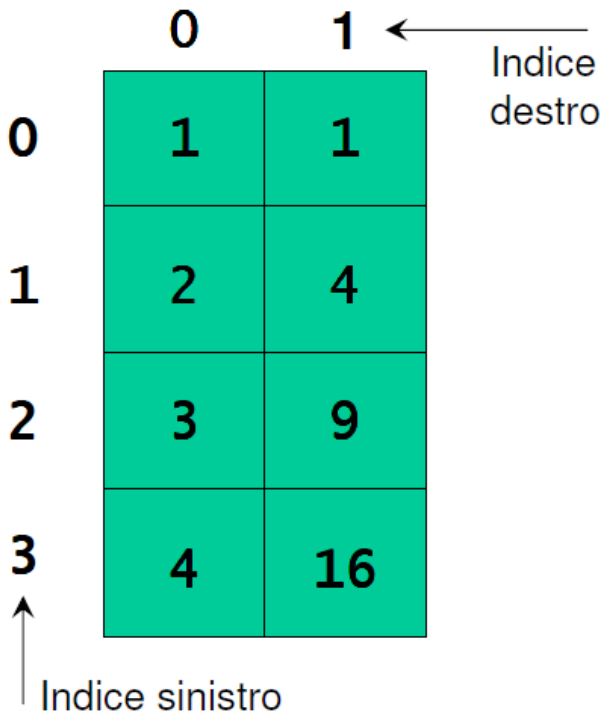
0FFC +[(2 * 4)+ 2=1034

Vettori bidimensionali (Matrici)

Gli array a più dimensioni vengono inizializzati allo stesso modo di quelli monodimensionali, ad esempio:

```
int quadrati[4][2] = {
1,1,
2,2,
3,9,
4,16
};
```

Inizializza un array chiamato quadrati con i numeri da 1 a 4 ed i loro quadrati.



Vettori bidimensionali (Matrici)

- Inizializzazione

  int b[2][2]={1,2,3,4};

  int b[2][2] = { { 1, 2 }, { 3, 4 } };

1	2
3	4

- Se non sono sufficienti, gli elementi non inizializzati sono posti a zero

  int b[2][2] = { { 1 }, { 3, 4 } };

1	0
3	4

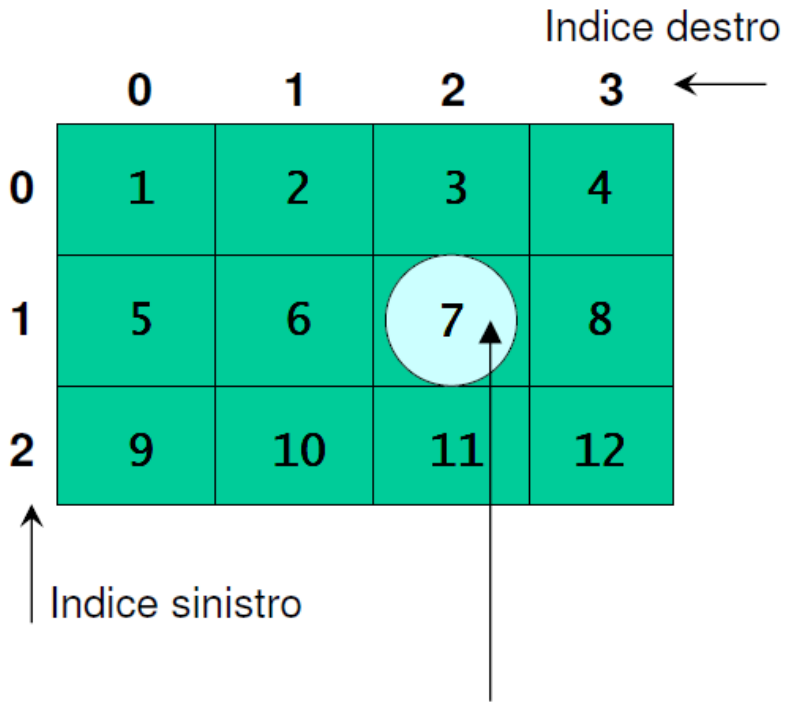
- Per riferirsi ad un elemento
  si specifica prima la riga, poi la colonna

  printf("%d", b[0][1]);

Esempio:

- cosa produce il seguente codice?

```
#include <stdio.h>
int main()
{
  int i,j,val[3][4];
  for(i=0;i<3;i++)
    {
     for(j=0;j<4;j++)
       {
        val[i][j]=(i*4)+j+1;
       }
     }
return 0;
}
```



[(indRig*NumCol)+indCol]=offset 0-11 `val[1][2]` 1*4+2 +1

Passaggio di matrici come parametri

- Quando passiamo una matrice ad una funzione, In ogni caso viene passato l'indirizzo dell'elemento di indice 0, associato al nome della matrice perche anch'essa è un vettore.
- Per poter calcolare l'offset corretto, la funzione deve conoscere il numero di colonne della matrice. Ricordiamo che il calcolo svolto è:

[(indRig*NumCol)+indCol]=offset

- Non possiamo specificare il parametro nella forma mat[][], come per i vettori, ma dobbiamo specificare il numero di colonne.

Esempio: void stampa(int mat[][5], int righe) { ... }

Esempio

```
Void stampa(double v[ ] [2], int,int)

int main(void)
{
 ...
double vet[2][2] = {
                    {1.1, 2.2},
                    {3.3, 4.4}
                    };

stampa(vet, 2,2);
...
}

void stampa(double v[ ][2], int r,int c)
 {
int i,j;
for (i = 0; i < r; i++)
for (j = 0; j < c;j++)
printf("%lf: %g\n", i, v[i][j]);
}
```

Prototipo della funzione

Dichiarazione ed inizializzazione

Chiamata alla funzione

Testata della funzione

Corpo della funzione

Array multidimensionali

- In C è possibile utilizzare array anche a più di due dimensioni la sintassi di utilizzo è sempre la stessa:

tipo nome [dim1] [dim2].....;

- Esempio di array a tre dimensioni: