

# Laboratorio di informatica

Ingegneria meccanica

Lezione 6 - 12 novembre 2007

1

## C: Array (1)

- Array individuato da **nome**, **tipo elementi** e **dimensioni**

`int v1[5]` ; dichiarazione array *monodimensionale* `v` di 5 interi `v1[0]`, `v1[1]`, `v1[2]`, `v1[3]`, `v1[4]`

- Array *bidimensionale*: elementi individuati da **due indici**

`int v2[2][3]` ; dichiarazione array *bidimensionale* `v2` di 6 interi `v2[0][0]`, `v2[0][1]`, `v2[0][2]`, `v2[1][0]`, `v2[1][1]`, `v2[1][2]`

2

## C: Array bidimensionali (2)

- `int v1[5] = { 0 , -1 , -2 , 3 , 4 } ;` dichiarazione con inizializzazione

- `int v2[2][3] = { { 1 , 2 , -1 } , { 0 , 3 , 7 } } ;`  
produce *inizializzazione per righe*

`v2[0][0] = 1 , v2[0][1] = 2 , v2[0][2] = -1 ,  
v2[1][0] = 0 , v2[1][1] = 3 , v2[1][2] = 7`

- Identica inizializzazione di `v2` prodotta con

`int v2[2][3] = { 1 , 2 , -1 , 0 , 3 , 7 } ;`

3

## C: Array bidimensionali (3)

- Se non vengono forniti valori per tutti gli elementi, si usano quelli forniti per *i primi indici* e per i mancanti si usa il valore `0`

- `int v2[2][3] = { 1 , 2 , -1 } ;`  
produce (*per righe*)

`v2[0][0] = 1 , v2[0][1] = 2 , v2[0][2] = -1 ,  
v2[1][0] = 0 , v2[1][1] = 0 , v2[1][2] = 0`

- `int v2[2][3] = { 0 } ;`

produce *inizializzazione a 0* di tutti gli elementi

4

## C: Array bidimensionali (4)

- Se non vengono forniti valori per tutti gli elementi di una riga si usano quelli forniti per *i primi indici* e per i mancanti si usa il valore 0

```
int v2[2][3] = { { 1 } , { 7 , -2 } } ;
```

produce (*inizializzazione per righe*)

```
v2[0][0] = 1 , v2[0][1] = 0 , v2[0][2] = 0 ,  
v2[1][0] = 7 , v2[1][1] = -2 , v2[1][2] = 0
```

5

## Scansione array monodimensionale

- Ciclo scansione di un array monodimensionale
- Esempi di utilizzazione: inizializzazione o aggiornamento valore elementi, ricerca elemento/elementi soddisfacenti condizioni assegnate, etc.
- ESEMPIO in C ( **v** array di **dim** elementi; **h** e **dim** tipo unsigned int )

```
for ( h = 0 ; h < dim ; h++ )  
{  
    ... /* analisi/modifica elemento v[ h ] */  
    ...  
}
```

6

## Ricerca elemento in un array

- ESEMPIO in C: Ricerca prima occorrenza elemento assegnato **w**
- **v** array di **dim** elementi double ; **h** , **dim** tipo unsigned int ; **found** tipo int ; **w** tipo double

```
...  
found = -1 ; /* rappresenta "elemento assente" */
```

```
for ( h = 0 ; ( h < dim ) && ( found == -1 ) ; h++ )  
{  
    if ( v[ h ] == w ) found = h ;  
}
```

```
...
```

7

## Scansione array bidimensionale

- ESEMPIO di scansione *per righe* in C ( **v** array di **dim1xdim2** elementi organizzati in **dim1** righe e **dim2** colonne ; **h** , **k** , **dim1** , **dim2** tipo unsigned int )

```
for ( h = 0 ; h < dim1 ; h++ ) {  
    for ( k = 0 ; k < dim2 ; k++ ) {  
        ... /* analisi elemento v[ h ][ k ] */  
        ...  
    } /* terminata scansione di una riga */  
} /* terminata scansione di tutte le righe */
```

8

## C: Ricerca valore massimo

```
int max = v[ 0 ][ 0 ] ;  
  
for ( h = 0 ; h < dim1 ; h++ ) {  
    for ( k = 0 ; k < dim2 ; k++ ) {  
        if ( v[ h ][ k ] > max )  
            max = v[ h ][ k ] ;  
    }  
}
```

9

## Ancora sugli array

- **Array monodimensionale**: collezione di **N** elementi omogenei (tutti dello stesso tipo) individuati da un indice intero (in C, variabile tra **0** e **N - 1**).  
Estensione: array *multidimensionale*
- **Attenzione**: *controllare* che i valori usati come indici di un array rientrino negli intervalli specificati dalle dimensioni dell'array. In C, non vi sono controlli automatici di questo tipo

10

## Esercizio (1)

Leggere matrice quadrata e verificare se la matrice è simmetrica  $a[i][j] = a[j][i]$  per ogni  $i, j$

Esempio

```
1 4 5  
4 3 2  
5 2 7
```

Letture matrice

```
...  
for ( h = 0 ; h < dim ; h++ )  
    for ( k = 0 ; k < dim ; k++ ) {  
        printf("Inserisci l'elemento [%d] [%d]: ", h, k);  
        scanf ("%d", &M[h][k]);  
    }
```

11

## Esercizio (2)

verifica matrice simmetrica

```
...  
simmetrica = 1;  
for ( h = 0 ; (h < dim) && (simmetrica == 1); h++ )  
    for ( k = 0; (k < dim) && (simmetrica == 1); k++ )  
        if ( M[h][k] != M[k][h] )  
            simmetrica=0;
```

12