

Laboratorio di informatica

Ingegneria meccanica

Esercitazione 9 - 28 novembre 2007

1

C: Passaggio di array a funzioni (1)

E' possibile passare un intero array ad una funzione

Esempio

```
int max ( int v[4] ) ;
```

...

```
int w[4] ;
```

...

```
a = max( w ) ;
```

In questo caso gli elementi di **w** possono essere modificati da **max**. Passaggio di un array avviene **per riferimento**: viene passato l'indirizzo del primo elemento dell'array

2

C: Passaggio di array a funzioni (2)

E' opzionale specificare la dimensione dell'array nella dichiarazione/definizione di una funzione che ha un un array monodimensionale nella **lista argomenti**,

Esempio

```
int max ( int v[ ] ) ;
```

In caso di array bidimensionale si deve specificare almeno la seconda dimensione

Esempio

```
min ( int v[ ][7] ) ;
```

3

Esercizio 1 (1/4)

- Progettare una funzione `leggiArray` che riceva come parametri:
 - un array `vect`
 - un intero `n`

legga da standard input `n` numeri interi e li memorizzi in `vect`.

4

Esercizio 1 (2/4)

- Progettare una funzione `mediaArray` che ricevi come parametri:
 - tre array di interi (`vA`, `vB`, `vC`),
 - un array di reali (`vD`)
 - la dimensione `n` degli array (uguale per tutti)memorizzi in cui ciascun elemento dell'array `vD` la media aritmetica degli elementi corrispondenti nei tre array `vA`, `vB`, `vC`
$$vD[i] = (vA[i] + vB[i] + vC[i]) / 3, \text{ per } i = 0, \dots, n - 1$$

ESEMPIO

Se `n=4`, `vA=[3,5,7,1]`, `vB=[2,1,0,12]`, `vC=[5,4,4,4]`
allora `vD = [3.33,3.33,3.66,5.66]`

5

Esercizio 1 (3/4)

Inserire le funzioni all'interno di un programma contenente quanto necessario per verificare se il comportamento delle funzioni è corretto.

Definire, tramite direttiva al preprocessore, il valore per la costante `DIM` (dimensione degli array)

6

Esercizio 1 (4/4)

Prototipi funzioni

```
void leggiArray( int v[], int n);
```

```
void mediaArray( int vA[] , int vB[] , int vC[] , float vD[], int n);
```

Chiamate funzioni

```
leggiArray(vett1,dim);
```

```
leggiArray(vett2,dim);
```

```
leggiArray(vett3,dim);
```

```
mediaArray(vett1, vett2, vett3, vett_media, dim);
```

7

Esercizio 2 (1/2)

Progettare una funzione che, ricevuti come parametri:

- un array bidimensionale `mA` di `m`x`n` interi (`m` righe, `n` colonne)
- un array `vB` di `n` interi
- un array `vC` di `m` interi
- i due interi `m` e `n`,

assegni, per `k = 0, 1, ..., m-1`, all'elemento `vC[k]` il valore $\sum_i mA[k][i] * vB[i]$, `i = 0, ..., n-1`

ESEMPIO:

Se `m=2`, `n=3`,

```
mA= [ 1, 2, 3 ]  
      [ 4, 5, 6 ]
```

```
vB = [ 2, 1, 0 ],
```

```
allora vC = [ 4, 13 ]
```

8

Esercizio 2 (2/2)

Inserire la funzione all'interno di un programma contenente quanto necessario per verificare se il comportamento della funzione è corretto

Le dimensioni massime degli array devono essere definite mediante direttive al preprocessore usando i valori **25** per entrambe le costanti **MAX_R** e **MAX_C**.

Attenzione: in caso di passaggio di array bidimensionali, nel prototipo della funzione si deve specificare almeno la seconda dimensione dell'array (**MAX_C** in questo caso)

9

Esercizio 3 (1/2)

Progettare una funzione `prodottoDiagonali` che, ricevuti come parametri:

- un array bidimensionale **mat** di $n \times n$ interi (che rappresenta una matrice quadrata),

- la dimensione n ,

stampi il prodotto della somma degli elementi della diagonale principale per la somma degli elementi della diagonale secondaria.

10

Esercizio 3 (2/2)

ESEMPIO:

```
Se n=3,  
mat= [ 1, 4, 5 ]  
      [ 7, 1, 2 ]  
      [ 4, 4, 2 ]
```

allora il prodotto della somma delle due diagonali corrisponde a $(1+1+2) * (4+1+5) = 40$

Inserire la funzione all'interno di un programma contenente quanto necessario per verificare se il comportamento della funzione è corretto
Definire, tramite direttiva al compilatore, il valore **10** per la costante **MAX_D** (numero massimo di righe e colonne)

11

Esercizio 4 (1/2)

Progettare una funzione che, ricevuti:

- un carattere **ch**;

- un array **vA** di reali

- un array **vB** di reali;

- un array **vC** di reali

- un intero **d** indicante la dimensione di **vA**, **vB**, **vC**, interpreti **ch** come l'operazione da eseguire tra **vA[k]** e **vB[k]** ($k=0,1,\dots,d-1$) per produrre il valore da assegnare a **vC[k]**.

Le operazioni valide sono: **somma** ('+'), **differenza** ('-'), **prodotto** ('*'), e **divisione** ('/').

In caso di operatore valido, la funzione restituirà un carattere indicante l'operazione effettuata. In caso di operatore non valido, la funzione restituirà il carattere '?' e non effettuerà alcuna operazione tra gli elementi di **vA** e **vB**

12

Esercizio 4 (2/2)

ESEMPI

Se $ch='+'$, $d=4$, $vA = [0.0, 1.0, 0.1, 1.1]$,
 $vB = [0.0, 1.0, 0.2, -1.7]$, verrà restituito '+' e
sarà $vC = [0.0, 2.0, 0.3, -0.6]$

Se $ch='%'$, $d=4$, $vA = [0.0, 1.0, 0.1, 1.1]$,
 $vB = [0.0, 1.0, 0.2, -1.7]$, verrà restituito '?' e
sarà $vC = [-, -, -, -]$

Inserire la funzione all'interno di un programma
contenente quanto necessario per verificare se il
comportamento della funzione è corretto.
Definire, tramite direttiva al compilatore, il valore
10 per la costante DIM (numero massimo di elementi
degli array)

13

Esercizio 5 (1/2)

Progettare una funzione che, ricevuti:

- un array bidimensionale **mat** di $m \times n$ interi (m righe, n colonne)
- un intero m
- un intero n

restituisca l'indice della colonna la cui somma
degli elementi è massima

14

Esercizio 5 (2/2)

Se $n=3, m=3$

mat = [1,2,3]
 [4,5,6]
 [7,8,9]

risulta che la colonna di indice 2 è la colonna la cui
somma degli elementi è massima (18).

Inserire la funzione all'interno di un programma
contenente quanto necessario per verificare se il
comportamento della funzione è corretto
Definire, tramite direttiva al preprocessore, il
valore 10 per le costanti **MAX_R** e **MAX_D**

15