Esercitazione Matrici e Struct

Informatica B, AA 2017/2018

Luca Cassano

luca.cassano@polimi.it

19 Ottobre 2017

Breve riassunto della puntata precedente

Breve riassunto della puntata precedente

- Matrici – array multidimensionali
- Acquisizione e stampa di una matrice
- Memorizzazione di array multidimensionali

```
#define R 100
#define C 50
int mat[R][C];
```

Acquisizione di una Matrice

```
int i, j;
for(i = 0; i < R; i++)
      for(j = 0; j < C ; j++)
      {
            printf("Inserire elemento posizione
                  [%d][%d]" , i+1 , j+1);
            scanf("%d", &mat[i][j]);
      }
```

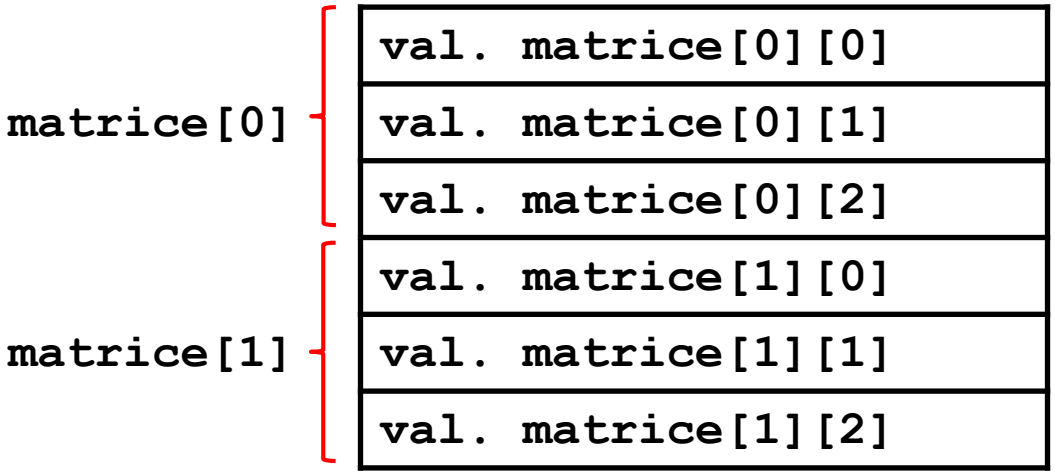
Stampa di una Matrice

```
for(i = 0; i < R; i++)
      {
            for(j = 0; j < C ; j++)
                printf("%5d" , mat[i][j]);
            printf("\n");
      }
```

Mappa di Memorizzazione di un array 2D

```
&matrice[i][j] == &matrice[0][0]+(i*Dim2)+j
```

```
int   matrice [2][3];
```



Indirizzi crescenti in memoria centrale

Breve riassunto della puntata precedente

- Strutture
- Allocazione di strutture in memoria
- Acquisizione e stampa dei campi di una struttura
- Assegnamento fra strutture
- Definizione di tipi definiti dall'utente

Dichiarazione di una struct, acquisizione e stampa

```
struct {
        char Nome[30];
        char Cognome[30];
        int Stipendio;
} dip1, dip2;

printf("\nInserire Nome: ");
scanf("%s", dip1.nome);
printf("\nInserire Cognome: ");
scanf("%s", dip1.cognome);
printf("\nInserire Stipendio: ");
scanf("%d", &dip1.stipendio);
printf("%s %s, guadagna %d",
                        dip1.nome,dip1.cognome,dip1.stipendio);
```

Assegnamento fra struct

```
dip2 = dip1;

dip2.stipendio += 10000;

printf("%s %s, guadagna %d",
                          dip2.nome,dip2.cognome,dip2.stipendio);
```

Definizione di Nuovi Tipi Strutturati

- Se si combina **`typedef`** con un costruttore **`struct`** o **array** i vantaggi diventano più evidenti.

```
typedef struct {
                   int giorno;
                   char mese[20];
                   int anno;
                 } Data;
```

Definizione di Nuovi Tipi Strutturati

- Quando si associa un nuovo tipo ad una struttura è possibile:
  1. dichiarare **altre strutture** (i.e., variabili del nuovo tipo)
  2. dichiarare **array** di strutture (i.e., array del nuovo tipo)
  3. utilizzare il nuovo tipo come **campo** di altre **strutture**
  4. utilizzare il nuovo tipo come **tipo base per nuovi tipi**

Definizione di Nuovi Tipi Strutturati

- Dichiarare **altre strutture** (i.e., variabili del nuovo tipo)

```
Data oggi, domani, dopoDomani;
```

Definizione di Nuovi Tipi Strutturati

- Dichiarare **altre strutture** (i.e., variabili del nuovo tipo)

```
Data oggi, domani, dopoDomani;
```

- Dichiarare **array** del **nuovo tipo** (i.e., array di strutture)

```
Data calendario[365];
Data settimana[7];
Data andataRitorno[2];
// popolare andataRitorno[0] per l'andata
andataRitorno[0].giorno = 12;
strcpy(andataRitorno[0].mese, "dicembre");
andataRitorno[0].anno = 2012;
// ritorno è come l'andata
andataRitorno[1] = andataRitorno[0];
// posticipo di 10 giorni il ritorno
andataRitorno[1].giorno += 10;
```

Definizione di Nuovi Tipi Strutturati

- Utilizzare il nuovo tipo come **campo** di altre **strutture**

```
struct {   char nome[30];
           char cognome[30];
           int stipendio;
           char codiceFiscale[16];
           Data dataDiNascita;} dip1;
```

Definizione di Nuovi Tipi Strutturati

- Utilizzare il nuovo tipo come **campo** di altre **strutture**

```
struct {   char nome[30];
           char cognome[30];
           int stipendio;
           char codiceFiscale[16];
           Data dataDiNascita;} dip1;
```

- Utilizzare il nuovo tipo come **tipo base per nuovi tipi**

```
typedef struct {char nome[30];
        char cognome[30];
        int stipendio;
        char codiceFiscale[16];
        Data dataDiNascita;
        } Dipendente;
```

ESERCIZI

Esercizio su Matrici 1

- Scrivere un programma che richiede l'inserimento di una matrice di interi mat di dimensioni **`M`** x **`N`** e di un intero **`n`**.
- Il programma conta il numero di occorrenze di **`n`** in ogni riga di **`M`**.

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;









}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;
        for(i = 0; i < M; ++i)
                for(j = 0; j < N, ++j)
                        scanf("%d", &mat[i][j]);









}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;
        for(i = 0; i < M; ++i)
                for(j = 0; j < N, ++j)
                        scanf("%d", &mat[i][j]);
        scanf("%d", &n);









        }
}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;
        for(i = 0; i < M; ++i)
                for(j = 0; j < N, ++j)
                        scanf("%d", &mat[i][j]);
        scanf("%d", &n);
        for(i = 0; i < M; ++i) {









        }
}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;
        for(i = 0; i < M; ++i)
                for(j = 0; j < N, ++j)
                        scanf("%d", &mat[i][j]);
        scanf("%d", &n);
        for(i = 0; i < M; ++i) {
                quanti = 0;









        }
}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
        int mat [M][N], n, i, j, quanti;
        for(i = 0; i < M; ++i)
                for(j = 0; j < N, ++j)
                        scanf("%d", &mat[i][j]);
        scanf("%d", &n);
        for(i = 0; i < M; ++i) {
                quanti = 0;
                for(j = 0; j < N, ++j)




        }
}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
	int mat [M][N], n, i, j, quanti;
	for(i = 0; i < M; ++i)
		for(j = 0; j < N, ++j)
			scanf("%d", &mat[i][j]);
	scanf("%d", &n);
	for(i = 0; i < M; ++i) {
		quanti = 0;
		for(j = 0; j < N, ++j)
			if(mat[i][j] == n)
				quanti++;


	}
}
```

Esercizio su Matrici 1

```
#include <stdio.h>
#define M 100
#define N 100

void main() {
	int mat [M][N], n, i, j, quanti;
	for(i = 0; i < M; ++i)
		for(j = 0; j < N, ++j)
			scanf("%d", &mat[i][j]);
	scanf("%d", &n);
	for(i = 0; i < M; ++i) {
		quanti = 0;
		for(j = 0; j < N, ++j)
			if(mat[i][j] == n)
				quanti++;
		printf("Alla riga %d ho trovato %d elementi uguali
a %d", i, quanti, n);
	}
}
```

Gara podistica

- Scrivere un programma che gestisca l'iscrizione degli atleti ad una gara podistica.
  - Alla gara possono iscriversi al più 10 atleti.
  - Ogni atleta ha un nome di al più 31 caratteri ed un numero di pettorale

Gara podistica 1

- Definire le strutture dati necessarie a risolvere il problema

Gara podistica 1

- Definire le strutture dati necessarie a risolvere il problema

```
#include <stdio.h>
#include <string.h>
const int MAX = 10;

typedef struct {
        char nome[32];
         int pettorale;
} Atleta;
```

Gara podistica 1

- Definire le strutture dati necessarie a risolvere il problema

```
#include <stdio.h>
#include <string.h>
const int MAX = 10;

typedef struct {
        char nome[32];
         int pettorale;
} Atleta;

typedef struct {
        Atleta atl[MAX];
        int quanti;
} Tabella;
```

Gara podistica 1

- Definire le strutture dati necessarie a risolvere il problema

```
#include <stdio.h>
#include <string.h>
const int MAX = 10;

typedef struct {
        char nome[32];
         int pettorale;
} Atleta;

typedef struct {
        Atleta atl[MAX];
        int quanti;
} Tabella;

void main() {
        Tabella tab;
        ……
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che inizializzi la tabella. Inizialmente la tabella e' vuota

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che inizializzi la tabella. Inizialmente la tabella e' vuota

```
.....
tab.quanti = 0;
.....
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che visualizza nome e numero degli atleti contenuti nella tabella

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che visualizza nome e numero degli atleti contenuti nella tabella

```
……
int i;
……
printf("%d atleti iscritti:\n", tab.quanti);
for(i=0; i<tab.quanti; i++) {
        printf("%s %d\n", tab.atl[i].nome, tab.atl[i].pettorale);
……
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

  - Se il nome e' gia' presente nella tabella, non viene effettuato l'inserimento.
  - Se il pettorale e' gia' presente nella tabella, non viene effettuato l'inserimento.
  - Se la tabella e' piena, non viene effettuato l'inserimento.

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
```

}

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
                if(strcmp(tab.atl[i].nome, nome)==0 ||
tab.atl[i].pettorale == pettorale)
                        flag = 1;
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
                if(strcmp(tab.atl[i].nome, nome)==0 ||
tab.atl[i].pettorale == pettorale)
                        flag = 1;
        if(flag == 0) {



        }
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
                if(strcmp(tab.atl[i].nome, nome)==0 ||
tab.atl[i].pettorale == pettorale)
                        flag = 1;
        if(flag == 0) {
                strcpy(tab.atl[tab.quanti].nome, nome);


        }
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
                if(strcmp(tab.atl[i].nome, nome)==0 ||
tab.atl[i].pettorale == pettorale)
                        flag = 1;
        if(flag == 0) {
                strcpy(tab.atl[tab.quanti].nome, nome);
                tab.atl[tab.quanti].pettorale = pettorale;

        }
}
```

Gara podistica 2

- Scrivere un frammento di codice da inserire nel main che aggiunge un nome e il corrispondente numero assegnato.

```
……
int i, flag = 0, pettorale;
char nome[32];
……
if(tab.quanti < MAX) {
        scanf("%s", nome);
        scanf("%d", pettorale);
        for(i=0; i<tab.quanti && flag == 0; i++)
                if(strcmp(tab.atl[i].nome, nome)==0 ||
tab.atl[i].pettorale == pettorale)
                        flag = 1;
        if(flag == 0) {
                strcpy(tab.atl[tab.quanti].nome, nome);
                tab.atl[tab.quanti].pettorale = pettorale;
                tab.quanti++;
        }
}
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
scanf("%s", nome);
scanf("%d", pettorale);
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
scanf("%s", nome);
scanf("%d", pettorale);
for(i=0; i<tab.quanti; i++) {









}
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
scanf("%s", nome);
scanf("%d", pettorale);
for(i=0; i<tab.quanti; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {




        }
}
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
scanf("%s", nome);
scanf("%d", pettorale);
for(i=0; i<tab.quanti; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {
                if(tab.quanti > 1) {
                        tab.atl[i] = tab.atl[tab.quanti-1];
                }

        }
}
```

Gara podistica 3

- Scrivere un frammento di codice da inserire nel main che elimina un atleta dalla tabella. L'atleta viene eliminato solo se presente nella tabella.

```
……
int i, pettorale;
char nome[32];
……
scanf("%s", nome);
scanf("%d", pettorale);
for(i=0; i<tab.quanti; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {
                if(tab.quanti > 1) {
                        tab.atl[i] = tab.atl[tab.quanti-1];
                }
                tab.quanti--;
        }
}
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
scanf("%s", nome);
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
scanf("%s", nome);
for(int i=0; i<tab.quanti && trovato == 0; i++) {




}
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
scanf("%s", nome);
for(int i=0; i<tab.quanti && trovato == 0; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {


        }
}
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
scanf("%s", nome);
for(int i=0; i<tab.quanti && trovato == 0; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {
                printf("%d", tab.atl[i].pettorale);
                trovato = 1;
        }
}
```

Gara podistica 4

- Scrivere un frammento di codice da inserire nel main che cerca un nome all'interno della tabella e stampa il pettorale.

```
……
int i, trovato = 0;
char nome[32];
……
scanf("%s", nome);
for(int i=0; i<tab.quanti && trovato == 0; i++) {
        if(strcmp(tab.atl[i].nome, nome) == 0) {
                printf("%d", tab.atl[i].pettorale);
                trovato = 1;
        }
}
if(trovato == 0)
        printf("Atleta non trovato!!!");
```