

## Array multidimensionali

Dott.ssa Veronica Rossano  
[rossano@di.uniba.it](mailto:rossano@di.uniba.it)  
<http://www.di.uniba.it/~rossano>

## Array multidimensionali...

- Gli array possono essere multidimensionali cioè avere due o più dimensioni
- Si utilizzano per rappresentare tabelle e/o matrici e qualsiasi altro oggetto a più dimensioni
- La sintassi è la seguente:

```
Tipo_valori nome_array [dim1] [dim2]...[dimN];
```

- `x[2][3]` /\* si riferisce all'elemento in terza riga quarta colonna \*/
- `retribuzione [gennaio][2006]` /\* si riferisce alla retribuzione di gennaio del 2006

Laboratorio di Programmazione - Veronica Rossano

2

## ...Array multidimensionali

ArrayMultidim-ValoriAssegnati.c

- L'inizializzazione di un array multidimensionale segue la stessa sintassi degli array monodimensionali

```
tipo_dato nome_vettore [N] [M]=  
{ {dato1, dato2,...,datoM},... {val1, val2,...,valM} };  
/*N volte*/  
  
double temp [7] [2]= { {10, 16.5}, {9, 12}, {-1, 5}, {5,  
9.5}, {4.5, 8}, {-2, 3}, {7, 12}};
```

## Esempio

- Scrivere un programma che richieda in input temperature minime e massime per ogni giorno della settimana e ne visualizzi la media delle temperature minime e la media delle temperature massime

## Rappresentazione grafica

|           | Min | Max  |
|-----------|-----|------|
| Lunedì    | 10  | 16.5 |
| Martedì   | 9   | 12   |
| Mercoledì | -1  | 5    |
| Giovedì   | 5   | 9.5  |
| Venerdì   | 4.5 | 8    |
| Sabato    | -2  | 3    |
| Domenica  | 7   | 12   |

```
#include <stdio.h>
#include <stdlib.h>
```

```
typedef enum
{min, max}
```

```
min_max;

typedef enum
{lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica}
settimana;
```

```
int
main (void)
{
    double temperatura[7][2];
    double media_min=0, media_max=0;
    settimana i;
    for(i=lunedì; i<=domenica; ++i)
    {
        printf("Inserire le temperature minima e massima del %d giorno", i+1);
        printf(" separate da uno spazio -->");
        scanf("%lf %lf", &temperatura[i][min], &temperatura[i][max]);
        media_min+=temperatura[i][min];
        media_max+=temperatura[i][max];
    }
    media_min/=7;
    media_max/=7;
    printf("La temperatura minima media della settimana e' stata : %.2f\n", media_min);
    printf("La temperatura massima media della settimana e' stata : %.2f\n", media_max);
    system("pause");
    return(0);
}
```

Tipi definiti  
dall'utente

ArrayMultidim-Temperatura.c

Dichiarazione dell'array  
multidimensionale

## Esercizio

TavolaPitagorica.c

- Scrivere un algoritmo e un programma che consenta di memorizzare e visualizzare la tavola pitagorica dei primi n numeri naturali. Consentire all'utente di decidere il valore di n (che comunque non dovrà essere superiore a 20). La visualizzazione dovrà essere nel formato:

```
1x1=1
1x2=2
1x3=3
1x4=4
1x5=5
...
```

## Le stringhe

Dott.ssa Veronica Rossano  
[rossano@di.uniba.it](mailto:rossano@di.uniba.it)  
<http://www.di.uniba.it/~rossano>

## Il tipo stringa

- In C una stringa (sequenza di caratteri) è implementata come un array
- La sintassi per la dichiarazione di una stringa è la seguente:

```
char nome_stringa[numero_caratteri];
```

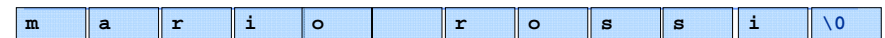
```
■ char nome[20];
```

## Inizializzazione

- La stringa in quanto array può essere inizializzata nei modi già discussi

```
char nome[20] = {'m','a','r','i','o',' ','r','o','s','s','i','\0'};  
char nome[20] = "mario rossi"
```

- La stringa è rappresentata in memoria come un insieme di celle contigue. Nell'ultima cella è memorizzato il carattere `\0` di fine stringa



## Array stringa multidimensionali

- È possibile dichiarare un array multidimensionale di caratteri
- Ogni elemento dell'array sarà una stringa di caratteri

```
char mesi [12] [10] = {"Gennaio", "Febbraio", "Marzo", "Aprile",  
"Maggio", "Giugno", "Luglio", "Agosto", "Settembre", "Ottobre",  
"Novembre", "Dicembre"};
```

**NB. In una assegnazione non è possibile andare a capo**  
Nella slide è riportato con questa formattazione solo per motivi di presentazione

## Stringa: Input e Output...

- Entrambe le funzioni `printf` e `scanf` possono utilizzare come argomenti una stringa semplicemente utilizzando il placeholder `%s`
- Nella `printf` è possibile specificare quanti caratteri minimo devono essere visualizzati
  - Scrivere `%8s` significa che **almeno** otto caratteri della stringa saranno visualizzati

## ...Stringa: Input e Output

- Nella Scanf non è necessario inserire l'operatore di indirizzo & in quanto il passaggio di un array avviene sempre per indirizzo
  - La scanf riceve una stringa in input con lo stesso processo adottato per gli input numerici
  - Ogni carattere digitato è memorizzato in celle contigue di memoria quando è digitato uno spazio, un tab o invio la scanf termina la memorizzazione

## Formattazione dell'output

Stringhe.c

```
char nome[20];
printf("Inserire un nome (max 20 caratteri)");
scanf("%s", nome);
printf("*****s*****\n", nome);
printf("*****4s*****\n", nome);
printf("*****20s*****\n", nome);
printf("*****-20s*****\n", nome);
```

G:\Lezioni\Programmazione\Prove\Array\stringhe.exe

```
Mario
*****Mario*****
*****Mario*****
*****          Mario*****
*****Mario          *****
Press any key to continue . . .
```

## Errore tipico

- La stringa **non è una variabile semplice** quindi non è possibile fare un assegnamento diretto del tipo

```
char nome[20];
nome = "Mario Rossi";
```

**errore in compilazione**

## La libreria <string.h>...

- La libreria string.h fornisce funzioni per rendere la gestione delle stringhe più agevole
  - strcpy(str\_dest, str\_sorg)
    - Consente di copiare la stringa sorgente nella stringa destinazione aggiungendo il carattere \0
  - strncpy(str\_dest, str\_sorg, n)
    - Consente di copiare i primi n caratteri della stringa sorgente nella stringa destinazione non aggiunge il carattere \0

## ...La libreria <string.h>...

- ❑ Strcat (str\_dest, str\_sorg)
  - Concatena la stringa sorgente alla stringa destinazione
- ❑ Strncat (str\_dest, str\_sorg, n)
  - Concatena i primi n caratteri della stringa sorgente alla stringa destinazione
- ❑ Strcmp(str1, str2)
  - Confronta due stringhe. Ritorna un **valore negativo** se str1 < str2, **0** se le stringhe sono uguali e un **valore positivo** se str1 > str2

## ...La libreria <string.h>...

- ❑ Strcmp(str1, str2, n)
  - Confronta i **primi n caratteri** delle due stringhe. Ritorna un **valore negativo** se str1 < str2, **0** se le stringhe sono uguali e un **valore positivo** se str1 > str2
- ❑ Strlen(str)
  - Ritorna il numero di caratteri della stringa non includendo il carattere di fine stringa

## ...La libreria <string.h>...

- ❑ Strtok(str, car)
  - Cerca nella stringa le sottostringhe delimitate dal carattere **car**, restituendone il puntatore al primo carattere della sottostringa. Le chiamate successive alla prima devono sostituire a str il valore NULL. La funzione restituisce NULL quando non trova più alcuna occorrenza del carattere car (token)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define LUNG 20
```

GestioneStringhe.c

```
int
main()
{
```

Dichiarazione e  
inizializzazione delle  
stringhe

```
char stringa1[LUNG] = "Mario Rossi";
char stringa2[LUNG] = "Studente Matr.33965";
char *punt;
printf("Stringa 1 -->%s\n", stringa1);
printf("Stringa 2 -->%s\n", stringa2);
```

Copia delle stringhe

```
printf("\n***** strcpy (stringa1, stringa2)*****\n");
strcpy(stringa1, stringa2);
printf("Stringa 1 -->%s\n", stringa1);
printf("Stringa 2 -->%s\n", stringa2);
```

Confronto tra stringhe

```
printf("\n***** strcmp (stringa1, stringa2)*****\n");
if (strcmp(stringa1, stringa2) == 0)
    printf("Le due stringhe sono uguali\n");
```

```

printf("\n***** strcpy (stringa1, stringa2, 5)*****\n");
strcpy(stringa1, "Mario Rossi"); /* prima di provare strcpy riportia
strcpy(stringa1, stringa2, 5);
printf("Stringa 1 -->%s\n", stringa1); /* da notare che strcpy non a
continua ad avere sempre lo

printf("Stringa 2 -->%s\n", stringa2);

if (strcmp(stringa1,stringa2)==0)
    printf("Le due stringhe sono uguali\n");
else if (strcmp(stringa1,stringa2)<0)
    printf("Stringa1 precede Stringa2\n");

printf("\n***** strcmp (stringa2, stringa1)*****\n");
if (strcmp(stringa2,stringa1)>0)
    printf("Stringa2 segue Stringa1\n");

printf("\n***** strncmp (stringa2, stringa1, 5)*****\n");
if (strncmp(stringa2,stringa1,5)==0)
    printf("I primi 5 caratteri delle stringhe sono uguali\n\n");

printf("\n***** strlen(stringa2)*****\n");
printf("La stringa 2 ha %d caratteri \n\n", strlen(stringa2));

```

Confronto dei primi 5  
caratteri

Lunghezza della stringa2

```

printf("\n***** strtok(stringa2, \",\") *****\n");
strcpy(stringa2, "Maria,Elena,Giovanni,Antonio,Ugo");
punt=strtok(stringa2,",");
while (punt!=NULL)
{
    printf("Il token estratto e' %s\n\n", punt);
    punt=strtok(NULL,",");
}

```

Sottostringhe della  
stringa 2 utilizzando il  
carattere ,

```

C:\eclipseC\workspace\Programmi Lezione 5\FunzioniStringhe\Debug\FunzioniStringhe.exe
Stringa 1 -->Mario Rossi
Stringa 2 -->Studente Matr.33965

***** strcpy (stringa1, stringa2)*****
Stringa 1 -->Studente Matr.33965
Stringa 2 -->Studente Matr.33965

***** strcmp (stringa1, stringa2)*****
Le due stringhe sono uguali

***** strcpy (stringa1, stringa2, 5)*****
Stringa 1 -->Stude Rossi
Stringa 2 -->Studente Matr.33965
Stringa 1 -->Stude
Stringa1 precede Stringa2

***** strcmp (stringa2, stringa1)*****

***** strncmp (stringa2, stringa1, 5)*****
I primi 5 caratteri delle stringhe sono uguali

***** strlen(stringa2)*****
La stringa 2 ha 19 caratteri

***** strtok(stringa2, ",")*****
Il token estratto e' Maria
Il token estratto e' Elena
Il token estratto e' Giovanni
Il token estratto e' Antonio
Il token estratto e' Ugo

Premere un tasto per continuare . . . _

```

## Overflow di stringhe

- Quando si assegna ad una stringa un valore che contenga più caratteri della lunghezza dichiarata della stringa si può incorrere in un errore logico
  - Il compilatore segnala solo un warning



## La libreria <ctype.h>...

- Nella gestione delle stringhe alcune volte può essere utile conoscere la natura di ciascun tipo di carattere
  - isalpha(ch)
    - Restituisce vero se ch è un carattere alfabetico
  - isdigit(ch)
    - Restituisce vero se ch è una cifra decimale
  - islower(ch) – isupper(ch)
    - Restituiscono vero se ch è rispettivamente un carattere minuscolo o maiuscolo

## ... La libreria <ctype.h>

- ispunct(ch)
  - Restituisce vero se ch è un carattere di punteggiatura
- isspace(ch)
  - Restituisce vero se ch è uno spazio
- Tolower(ch) – toupper(ch)
  - Convertono ch da un carattere maiuscolo in minuscolo e viceversa

### GestioneCaratteri.c

```
main()
{
    char stringa[LUNG] = "Studente Matr.33965";
    char ch;
    if (isalpha(stringa[0]))
        printf("\nIl carattere --> %c e' un carattere alfabetico", stringa[0]);
    if (isdigit(stringa[14]))
        printf("\nIl carattere --> %c e' un numero", stringa[14]);
    if (islower(stringa[10]))
        printf("\nIl carattere --> %c e' un carattere minuscolo", stringa[10]);
    ch=toupper(stringa[10]);
    printf("\nIl carattere e' trasformato in maiuscolo --> %c", ch);
    if (isupper(stringa[9]))
        printf("\nIl carattere --> %c e' un carattere MAIUSCOLO", stringa[9]);
    ch=tolower(stringa[9]);
    printf("\nIl carattere e' trasformato in minuscolo --> %c", ch);
    if (ispunct(stringa[13]))
        printf("\nIl carattere --> %c e' un carattere di punteggiatura", stringa[13]);
    if (isspace(stringa[8]))
        printf("\nIl carattere --> %c e' uno spazio\n", stringa[8]);

    system("pause");
}
```

```
C:\Documents and Settings\ssis\Desktop\Prove\Array e Stringhe\Gestione caratteri.exe
Il carattere --> S e' un carattere alfabetico
Il carattere --> 3 e' un numero
Il carattere --> a e' un carattere minuscolo
Il carattere e' trasformato in maiuscolo --> A
Il carattere --> M e' un carattere MAIUSCOLO
Il carattere e' trasformato in minuscolo --> m
Il carattere --> . e' un carattere di punteggiatura
Il carattere -->   e' uno spazio
Premere un tasto per continuare . . .
```

## Esercizio

AnalisiStringa.c

- Scrivere l'algoritmo e il programma data una stringa in input ne analizzi i singoli caratteri restituendo il numero di caratteri alfabetici e, di questi, specifichi quanti sono maiuscoli e quanti sono minuscoli, il numero di cifre e il numero di segni di punteggiatura