

ARRAY  
MULTIDIMENSIONALI  
MATRICI

# INTRODUZIONE

Le celle di un array possono essere di qualunque tipo, anche di tipo array.

Questo permette di avere array multidimensionali.

Utilizzando nella dichiarazione due coppie di parentesi si ottengono array bidimensionali, tre coppie array tridimensionali, e così via.

# esempi

- `int a[100]`      array monodimensionale
  - `int b[4][6]`      array bidimensionale
  - `int c[3][4][8]`    array tridimensionale
- e così via. Ogni coppia di parentesi nella dichiarazione aggiunge una dimensione.

# Array bidimensionale

```
int A[3][5];
```

A è un vettore di vettore. Ha 3 vettori, linee, composti da 5 celle, colonne.

	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4
Riga 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]
Riga 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[1][4]
Riga 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]

# utilizzo

```
int somma( int a[][5] ){  
    int i, k, s=0;  
  
    for(i=0;i<3; ++i)  
        for(k=0;k<5;++k)  
            s+=a[i][k];  
  
    return s;  
}
```

Nella dichiarazione tra parentesi:  
int a[][5]

È equivalente

int a[3][5]

Il compilatore non usa la costante  
3. Può servire solo a chi legge il  
programma.

# Dichiarazione con inizializzazione

```
int a[2][3] = { 1, 2, 3, 4, 5, 6};
```

```
int b[2][3] = { {1, 2, 3} , {4, 5, 6} };
```

```
int c[ ][3] = { {1, 2, 3} , {4, 5, 6} };
```

```
int d[2][3] = { 0 }
```

# #define

#define permette di definire costanti. E' un comando per il precompilatore, che sostituirà ad una sequenza di caratteri una nuova sequenza.

```
#define DIM 100
```

Il precompilatore sostituirà alla sequenza DIM la sequenza 100

# typedef

typedef permette di assegnare un nome, un identificatore, ad un tipo.

Il nuovo tipo così definito può essere utilizzato nelle dichiarazioni, rendendo più leggibile il sorgente.

```
typedef unsigned long size_t;
```

```
typedef long base;
```

```
typedef long altezza;
```

# Utilizzo di typedef

```
#define DIM 100
```

```
typedef double scalare;
```

```
typedef scalare vettore[DIM];
```

```
typedef vettore matrice[DIM];
```

matrice è un nuovo tipo. E' un vettore bidimensionale 100x100 di double.

Matrice a;

a è una variabile di tipo matrice

# Esempi: operazioni sul dominio

```
void add( vettore x, vettore y, vettore z) {  
    int i;  
    for(i=0; i<DIM; ++i)  
        z[i]=x[i]+y[i];  
}
```

# Esempi: operazioni sul dominio

```
scalare prodottoScalare( vettore x, vettore y){  
    scalare s = 0.0;  
    int i;  
    for(i=0; i<DIM; ++i)  
        s += x[i] * y[i];  
    return s;  
}
```

# Esempi: operazioni sul dominio

```
void moltiplica( matrice x, matrice y, matrice z){
    int i, j, k;
    for(i=0; i<DIM; ++i)
        for(j=0; j<DIM; ++j) {
            z[i][j] = 0;
            for(k=0; k<DIM; ++k)
                z[i][j] += a[i][k] * b[k][j];
        }
}
```

# Esempio.

Si supponga di avere a disposizione un vettore contenente l'elenco degli alunni iscritti al Politecnico, con la relativa data di nascita. Ogni nome è separato dalla data dal carattere ':'. La data è espressa come ggmmaaaa.

Risolvere i seguenti problemi:

1. Scrivere una funzione che restituisce la data di nascita di un determinato studente.
2. Scrivere una funzione che elimina tutti gli studenti nati in una determinata data.
3. Scrivere una funzione che conta il numero degli studenti nati in ogni mese dell'anno.
4. Scrivere una funzione che determina gli studenti più vecchi.