

# GLI ARRAY



# Gli array

Gli array sono strutture dati statiche, di tipo sequenziale, che consentono la memorizzazione e la gestione di uno o più dati omogenei (dello stesso tipo) raggiungibili per mezzo di un indice.

# Gli array

Per definire un array è necessario fornire:

- **Nome**
  - **Tipo**
  - **Dimensione**
- Come per le altre variabili**
- Dopo la dichiarazione della variabile, non può essere modificata**

## Gli array bidimensionali

Gli array bidimensionali o matrici sono quegli array per la cui definizione è sufficiente fornire due valori per la dimensione.

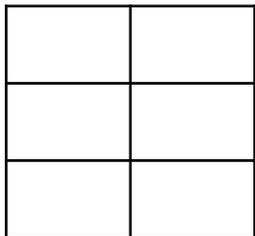
```
int m [5][2];
```


# Gli array bidimensionali

Anche nelle matrici i dati vengono memorizzati in modo sequenziale per facilitare l'accesso.

Data una matrice di interi con N righe ed M colonne vengono allocate NxM posizioni di memoria adatte ad accogliere interi.

```
int m [3][2];
```



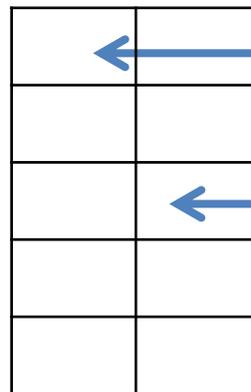
**E' indispensabile specificare il numero di colonne per sapere quando stanno iniziando i dati della riga successiva.**

# Elementi di una matrice

Per raggiungere gli elementi di una matrice è necessario precisare il nome della matrice e le coordinate dell'elemento desiderato.

In C il primo elemento di una matrice si trova in posizione 0,0.

~~m[i,j]~~    m[i][j]



m[0][0]

m[2][1]

Dati singoli che possono essere usati in espressioni o assegnazioni

Indici (numero o espressione intera positiva)

# Assegnazione valori a matrice

Per assegnare i valori ad una matrice è possibile:

## ❖ Assegnazione diretta

```
int m[2][2]={{7,9},{10,3}};
```

7	9
10	3

```
int m[2][2]={7,9,10,3};
```

```
int m[][2]={{7,9},{10,3}};
```

~~int m[2][2];~~

```
int m[2][2]={{3,1}};
```

3	1
0	0

```
int m[2][2]={{0}};
```

0	0
0	0

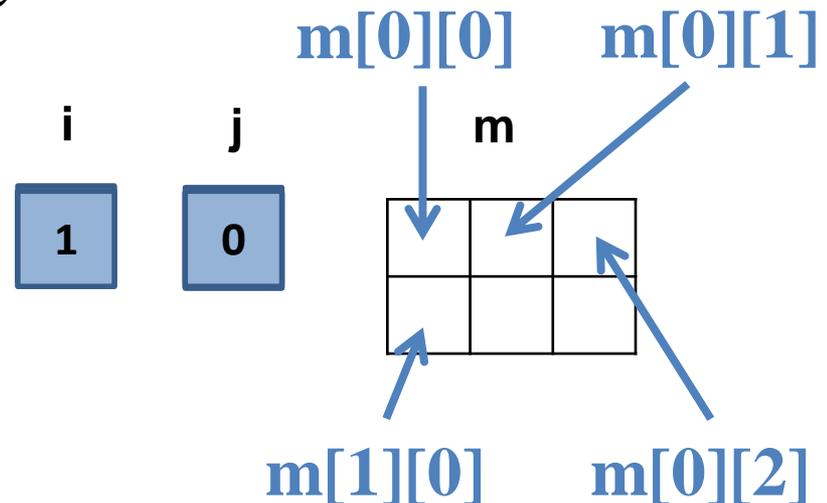
# Assegnazione valori a matrice

Per assegnare i valori ad una matrice è possibile:

- ❖ Assegnazione diretta
- ❖ Acquisizione da tastiera

E' necessario assegnare i valori alle singole componenti della matrice

```
int i,j, m[2][3];  
for(i=0;i<2;i++)  
    for(j=0;j<3;j++)  
        cin >> m[i][j];
```



# Assegnazione valori a matrice

Per assegnare i valori ad una matrice è possibile:

- ❖ Assegnazione diretta
- ❖ Acquisito da tastiera
- ❖ Successivo ad un calcolo

```
int i, j, m[3][2];
```

```
for(i=0;i<3;i++)
```

```
    for(j=0;j<2;j++)
```

```
        m[i][j]=2*i+3*j;
```

0	3
2	5
4	7

## Visualizzazione dei valori di una matrice

Per visualizzare i valori memorizzati in una matrice è necessario scorrere le singole componenti e visualizzarle:

```
for(i=0;i<N;i++)  
    for(j=0;j<M;j++)  
        cout << m[i][j];
```

In questo modo i valori verranno visualizzati in fila.  
E se volessi visualizzarli in forma tabulare?

## Visualizzazione dei valori di una matrice

Per visualizzare i valori memorizzati in una matrice è necessario scorrere le singole componenti e visualizzarle:

```
for(i=0;i<N;i++)  
    {for(j=0;j<M;j++)  
        cout << m[i][j];  
    cout << endl;}
```

**N è il numero di righe ed M il numero di colonne della matrice, impostati in fase di dichiarazione.**

**Può essere utile definire N ed M con una #define.**

## Esercizio

Data una matrice 3x2, calcolare e memorizzare la somma degli elementi di ciascuna riga:

```
main()
{int i=0, j=0;
 int m[3][2], v[3]={0};
 ...
 for(i=0;i<3;i++)
     for(j=0;j<2;j++)
         v[i]=v[i]+m[i][j];
```

i	j	m		v
0	1	5	3	8
1	1	2	5	7
2	1	4	7	11

## Passaggio di matrice a funzione

`void fun(int a[10][5]);` La prima dimensione può essere omessa ma non la seconda.

`main()` `fun(int a[10][5])`

`{int m[10][5];` `{...}`

...

`fun(m);` ← Anche le matrici vengono sempre passate per riferimento, quindi se non voglio apportare modifiche uso il qualificatore `const`.

`...}`

# **Gli array multidimensionali**

Il C/C++ consente di definire array con qualunque dimensione.