Titolo unità didattica: Strutture dati: array [07]

Titolo modulo : Function in C per problemi di base con array 2D [11-C]

Array 2D : mappa di memorizzazione e utilizzo

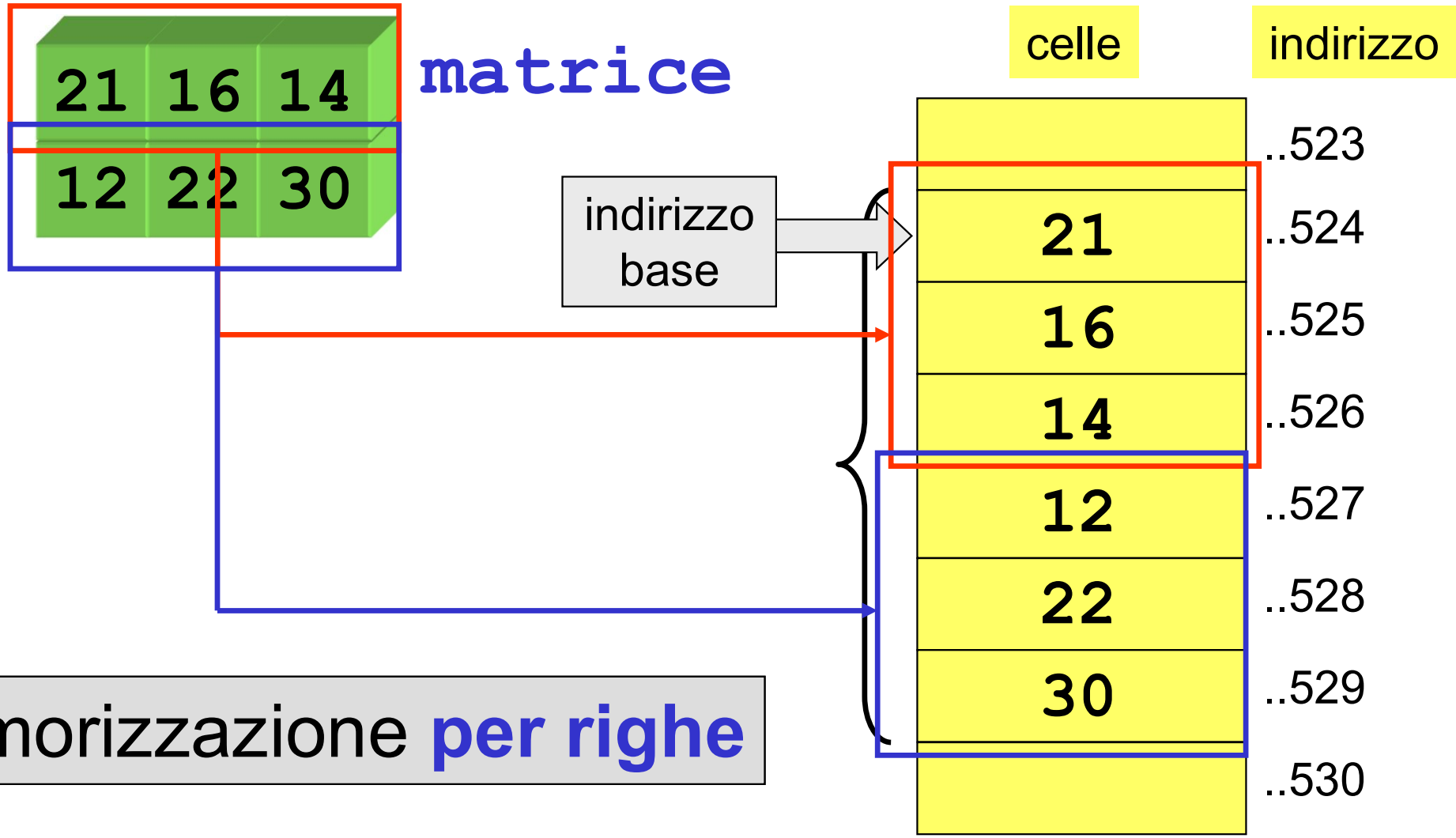
Argomenti trattati:

- ✓ mappa di memorizzazione di array 2D in C
- ✓ passaggio di array 2D a una function C
- ✓ function in C per la somma, il massimo e la ricerca in array 2D
- ✓ function in C per problemi di base su porzioni di array 2D
- ✓ function in C per il trattamento elementare di immagini

Prerequisiti richiesti: AP-07-08-T

rappresentazione di array 2D

```
int matrice[2][3]= {{21,16,14},
                    {12,22,30}};
```

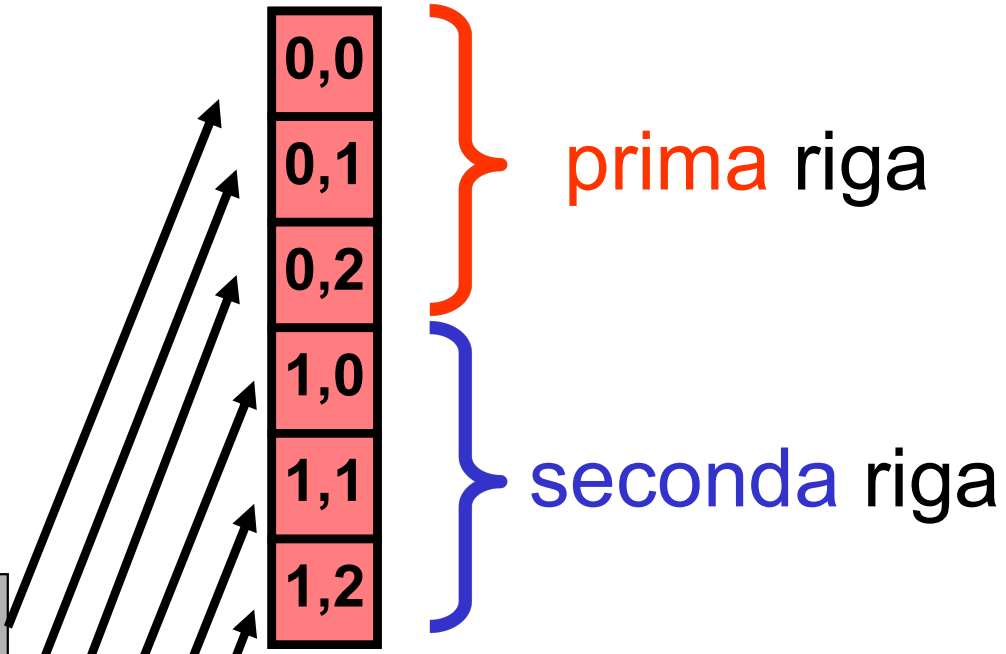


memorizzazione **per righe**

rappresentazione di array 2D

0,0	0,1	0,2
1,0	1,1	1,2

array 2D 2x3



indirizzo primo elemento
+ 1
+ 2
+ 3
+ 4
+ 5

indirizzo base di un array 2D

`&matrice[0][0]`

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

indirizzo base di un array 2D

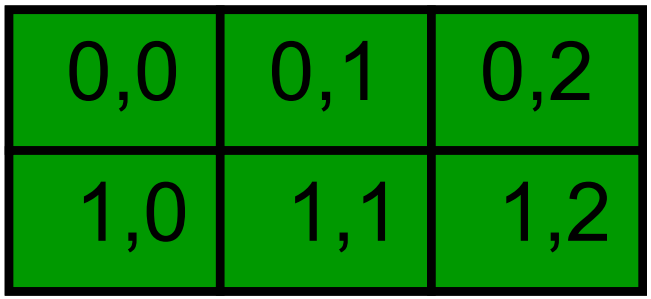
```
&matrice[0][0]
```

il nome di un array 2D **non** è un puntatore costante all'indirizzo base dell'array 2D

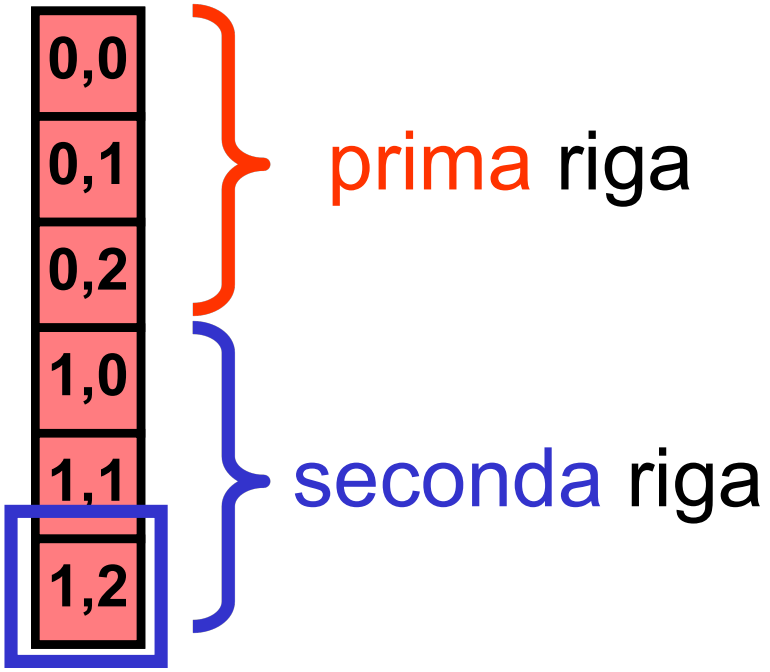
`matrice`	è equivalente a	`&matrice[0]`

è un puntatore all'indirizzo base di un array di 2 puntatori, ognuno dei quali punta a un array di 3 `int`

un array 2D può essere considerato un **array di puntatori ad array**

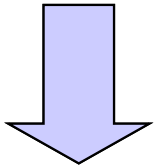


array 2D 2x3



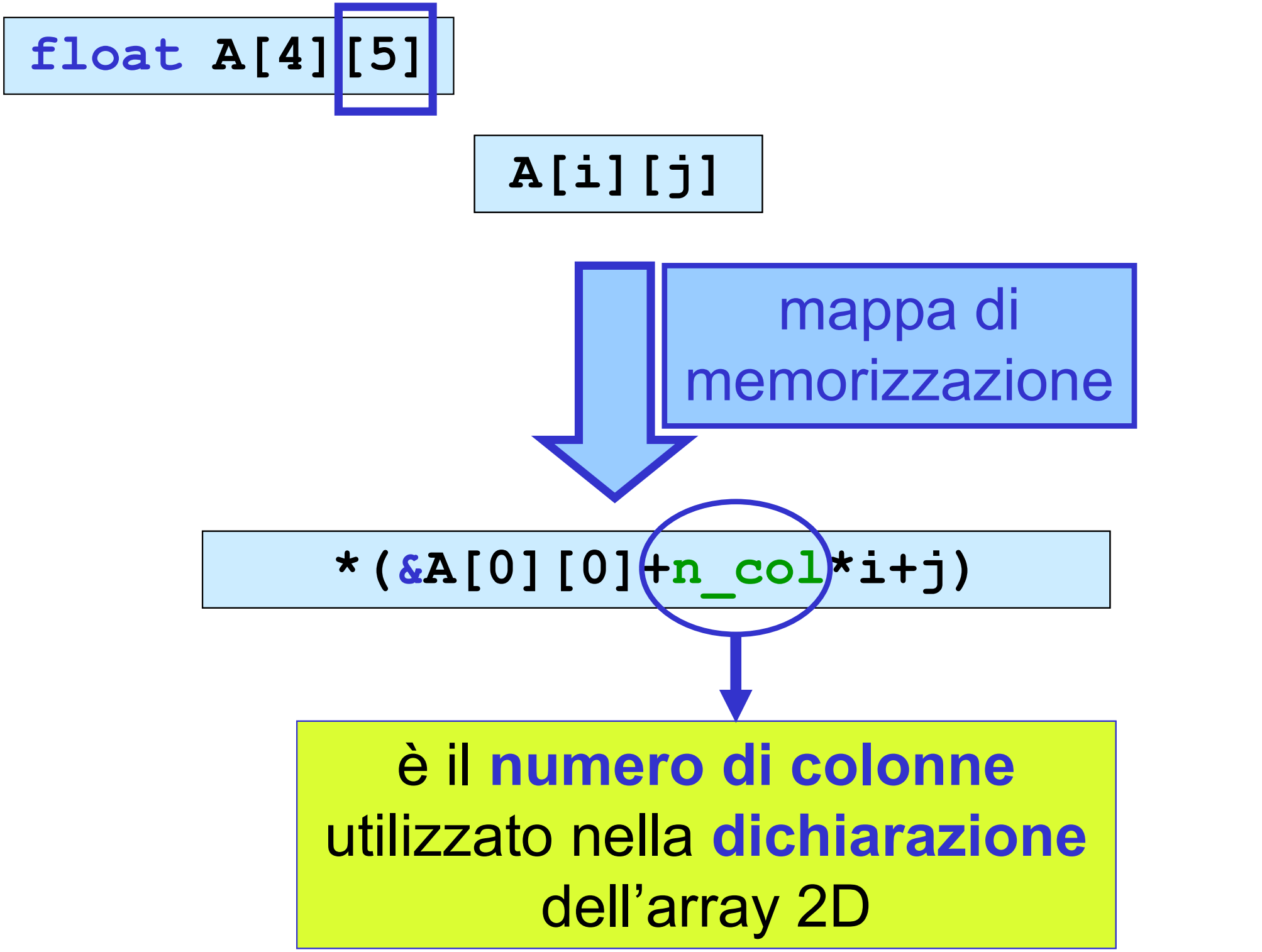

```
int matrice[2][3];
```

```
matrice[1][2]
```



```
*(&matrice[0][0]+3*1+2)
```

```
*(indirizzo base  +5)
```


float A[4][5]
A[i][j]
mappa di memorizzazione
*(&A[0][0]+n_col*i+j)
è il numero di colonne utilizzato nella dichiarazione dell'array 2D

```
float A[n_rig][n_col]
```

```
A[i][j]
```

trasformazione eseguita dal **compilatore**

mappa di memorizzazione

```
*(&A[0][0]+n_col*i+j)
```

è il **numero di colonne** utilizzato nella **dichiarazione** dell'array 2D

un elemento di un array 2D può essere denotato

- ✓ attraverso due **indici** (notazione **standard**)
- ✓ dereferenziando la mappa di memorizzazione (notazione a **puntatore**)

```
int M[n_rig][n_col];
```

```
M[0][0]
M[i][j]
M[i][j]
M[i][j]
```

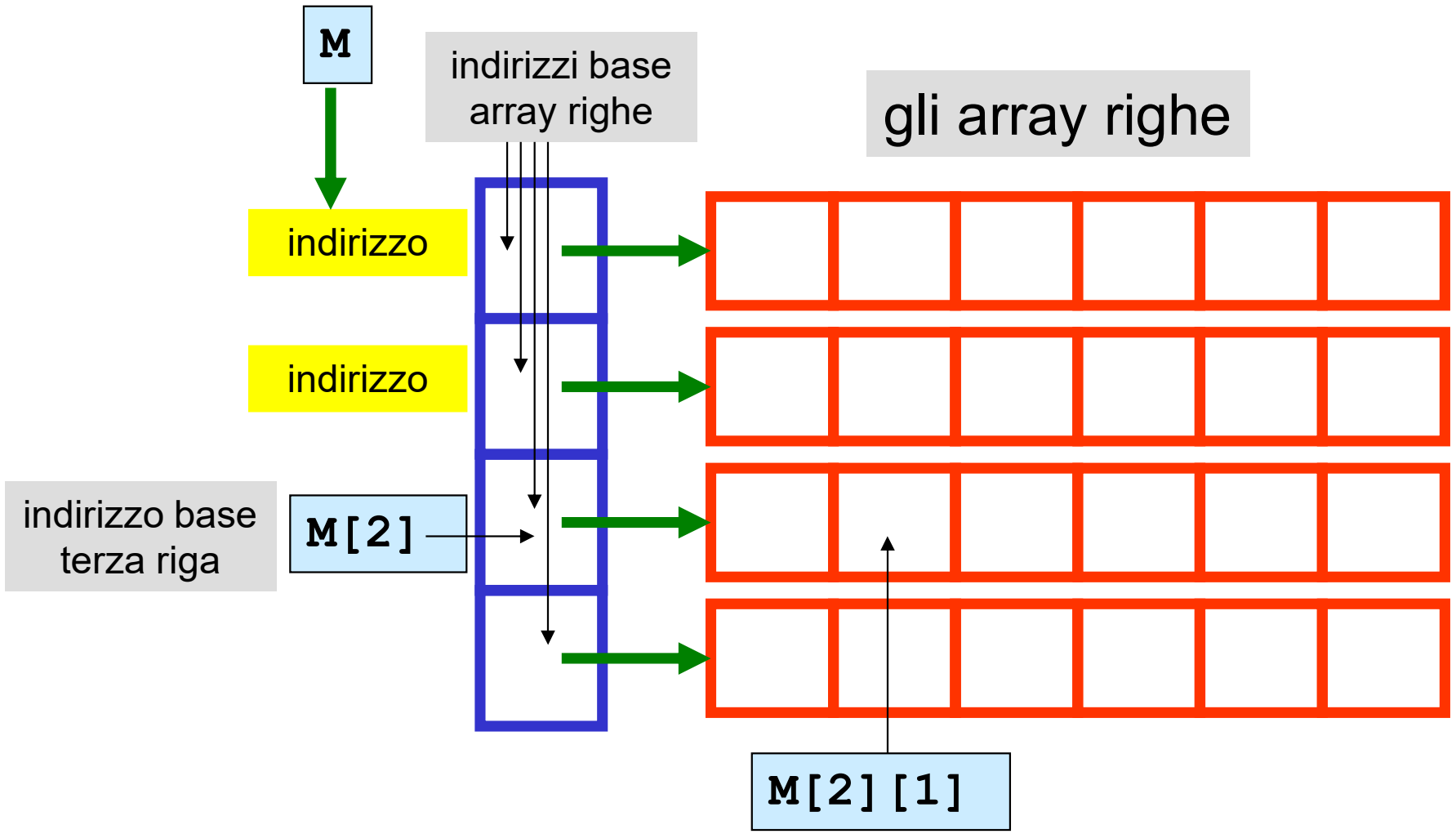
notazione standard

```
*(&M[0][0])
*(&M[0][0]+n_col*i+j)
*(M[i]+j)
(*(M+i))[j]
```

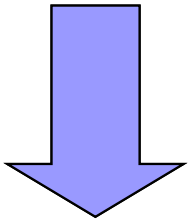
notazione a puntatore

un array 2D può essere considerato un **array di puntatori ad array**

```
int M[4][6];
```



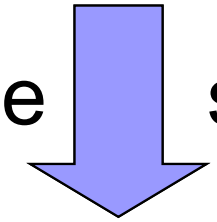
passaggio di un array 2D a una function



nella function deve essere **esplicitamente specificato** il **numero delle colonne di un parametro array,** affinché il compilatore possa generare la corretta mappa di memorizzazione

non è necessario conoscere il **numero delle righe**

passaggio di un array 2D a una function

allocazione  statica

quando un array 2D appare come parametro nell'intestazione di una function, il **numero delle colonne** deve essere **esattamente uguale** al valore specificato per il numero delle colonne nella dichiarazione dell'**argomento nel chiamante**

esercizio

realizzare una function C che visualizza gli elementi di un array 2D

```
/* visualizzazione degli elementi di un array 2D int
  - notazione standard                                  */
void  visualizza_a2DI(int a[][100], int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
   {
     for (j=0;j<m;j++)
         printf("%5d", a[i][j]);
     printf("\n");
   }
}
```

- size colonne nella dichiarazione
- numero effettivo righe utilizzate
- numero effettivo di colonne utilizzate

```
void visualizza_a2DI (int (*a)[100], int n,int m)
```

esercizio

realizzare una function C che visualizza gli elementi di un array 2D

```
/* visualizzazione degli elementi di un array 2D int
  - notazione a puntatore                              */
void visualizza_a2DIp(int *pa,int n_col, int n, int m)
{
 int i,j;
 for (i=0;i<n;i++)
   {
     for (j=0;j<m;j++)
         printf("%5d", *(pa+n_col*i+j));
     printf("\n");
   }
}
```

size colonne nella dichiarazione

numero effettivo di righe utilizzate

numero effettivo di colonne utilizzate

```c
#include <stdio.h>
void visualizza_a2DIp(int *,int,int,int);
void main()
{
 int a[2][3]={{1,2,3},{4,5,6}};
 int b[4][4]={{10,20,30,40},{50,60,70,80},{100,200,300,400},{500,600,700,800}};
 int c[3][2]={{9,9},{7,6},{5,4}} ;
  visualizza_a2DIp(&a[0][0], 3,  2, 3);
  visualizza_a2DIp(&b[0][0], 4,  4, 4);
  visualizza_a2DIp(&c[0][0], 2,  3, 2);
}
void visualizza_a2DIp(int *pa,int n_col, int n, int m)
{
 int i,j;
 for (i=0;i<n;i++)
   {
     for (j=0;j<m;j++)
         printf("%5d", *(pa+n_col*i+j));
     printf("\n");
   }
}
```

```c
#include <stdio.h>
void visualizza_a2DIp(int *,int,int,int);
void main(){
 int a[2][3]={{1,2,3},{4,5,6}};
 int b[4][4]={{10,20,30,40},{50,60,70,80},{100,200,300,400},{500,600,700,800}};
 int c[3][2]={{9,9},{7,6},{5,4}} ;
  visualizza_a2DIp(&a[0][0],3,  2, 2);
  visualizza_a2DIp(&b[0][0],4,  3, 3);
  visualizza_a2DIp(&c[0][0],2,  3, 1);}
```

esercizio

realizzare una function C che calcola la somma degli elementi di un array 2D

```
/* somma degli elementi di un array 2D int
  - notazione standard                                    */
int  somma_array2DI(int a[][100], int n, int m)
{
 int s=0,i,j;
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
         s = s + a[i][j];
 return s;
}
```

esercizio

realizzare una function C che calcola la somma degli elementi di un array 2D

```
/* somma degli elementi di un array 2D int
  - notazione a puntatore                    */
int somma_array2DI(int *pa, int n_col, int n,int m)
{
 int s=0,i,j;
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
         s = s + *(pa+n_col*i+j);
 return s;
}
```

esercizio

realizzare un main che chiama la function C che calcola la somma degli elementi di un array 2D

```
#include <stdio.h>
int somma_array2DI(int [][100], int, int);
void visualizza_a2DI(int [][100], int, int );
void main ()
{
 int a[100][100],i,j,somma,n,m;
 printf("inserire il numero di righe (<=100):");
 scanf("%d",&n);
 printf("inserire il numero di colonne (<=100):");
 scanf("%d",&m);
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
       scanf("%d",&a[i][j]);
 somma = somma_array2DI(a,n,m);
 printf ("somma degli elementi: %d \n",somma);
 printf ("l'array e'\n");
 visualizza_a2DI(a,n,m);
}
```

la somma degli elementi di un array 2D può essere calcolata usando la function **somma_arrayI** per gli array 1D

```
#include <stdio.h>
void visualizza_a2DIp(int *,int,int,int);
int somma_arrayI(int [],int);
void main ()
{
 int a[3][3]={{2,1,3},{6,9,1},{6,5,7}};
 int i,j,somma,n=3,m=3;
 somma = somma_arrayI (&a[0][0], n*m);
 printf ("somma degli elementi: %d \n",somma);
 printf ("l'array e'\n");
 visualizza_a2DIp(&a[0][0],3,n,m);
}
```

attenzione: gli elementi di un array 2D non sono memorizzati in celle consecutive, nel caso di *utilizzo parziale*

la somma degli elementi di un array 2D può essere calcolata usando la function **somma_arrayI** per gli array 1D

```
#include <stdio.h>
int somma_arrayI(int [],int
void visualizza_a2DI(int []
void main ()
{
 int a[100][100],i,j,somma,n,m;
 printf("inserire il numero di righe (<=100):");
 scanf("%d",&n);
 printf("inserire il numero di colonne (<=100):");
 scanf("%d",&m);
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
       scanf("%d",&a[i][j]);
 somma = somma_arrayI(&a[0][0], n*m);
 printf ("somma degli elementi: %d \n",somma);
 printf ("l'array e'\n");
 visualizza_a2DI(a,n,m);
}
```

attenzione: gli elementi di un array 2D non sono memorizzati in celle consecutive, nel caso di *utilizzo parziale (***n**,**m** < 100*)*

esercizio

realizzare una function C che determina il massimo elemento di un array 2D

```
/* massimo tra gli elementi di un array 2D int
  - notazione standard                          */
int massimo_array2DI(int a[][100],int n,int m)
{
 int max,i,j;
 max = a[0][0];
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
          if(a[i][j] > max)
              max = a[i][j];
 return max;
}
```

esercizio

realizzare una function C che determina il massimo elemento di un array 2D

```
/* massimo tra gli elementi di un array 2D int
  - notazione a puntatore                        */
int massimo_array2DIp(int *pa,int n_col,int n,int m)
{
 int max, i,j;
 max = *pa;
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
          if(*(pa+n_col*i+j) > max)
              max = *(pa+n_col*i+j);
 return max;
}
```

esercizio

realizzare un main che chiama la function C che determina il massimo elemento di un array 2D

```
#include <stdio.h>
int massimo_array2DIp(int *,int,int,int);
void visualizza_a2DI(int [][100],int,int);
void main ()
{
 int a[100][100],massimo,n,m;
 printf("inserire il numero di righe (<=100):");
 scanf("%d",&n);
 printf("inserire il numero di colonne (<=100):");
 scanf("%d",&m);
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
       scanf("%d",&a[i][j]);
 massimo = massimo_array2DIp(&a[0][0],100,n,m);
 printf ("il massimo tra gli elementi: %d \n",massimo);
 printf ("l'array e'\n");
 visualizza_a2DI(a,n,m);
}
```

esercizio

realizzare una function C che determina l'appartenenza di una chiave a un array 2D

```
/* function che determina l'appartenenza di un
       dato(chiave) a un array 2D
   --algoritmo di ricerca sequenziale per array 2D--
*/
int appartiene_a2DF(float chiave,float a[][100],int n,int m)
 {
  int i,j;
  for(i=0; i<n; i++)
      for(j=0; j<m; j++)
          if (chiave == a[i][j])
              return 1;
  return 0;
 }
```

esercizi

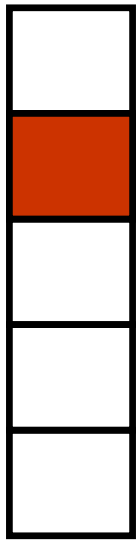
realizzare un main C che legge da tastiera un array 2D di tipo `double` (insieme con il numero effettivo di righe e di colonne) e calcola, visualizzando su schermo:

- ✓ la media di ogni riga
- ✓ la media di ogni colonna

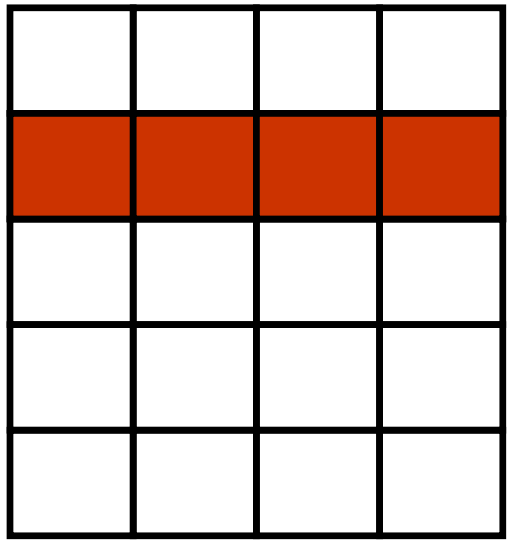
main C per il calcolo della media di ogni riga di un array 2D

dati di input: valori dell'array 2D `A[][],n,m`
dati di output: le `n` medie: array `media_riga[]`

`media_riga`

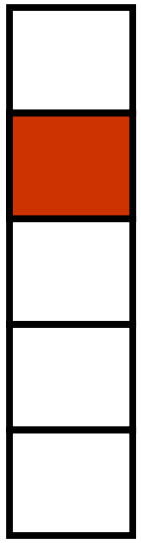


`A`

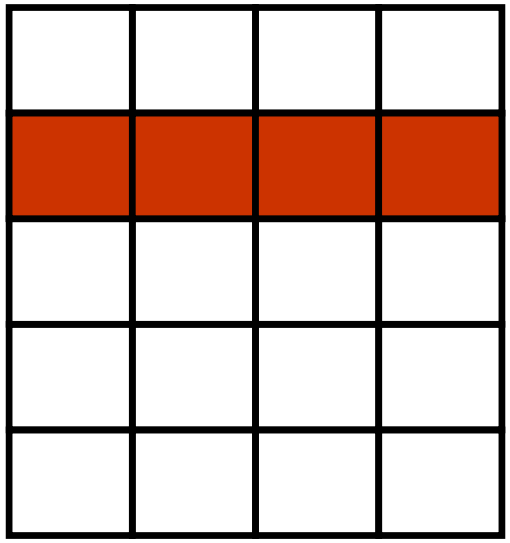


```
for (i=0;i<n;i++)
    determinare il valore di media_riga[i]);
```

media_riga

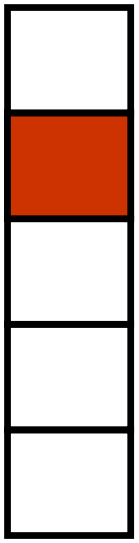


A

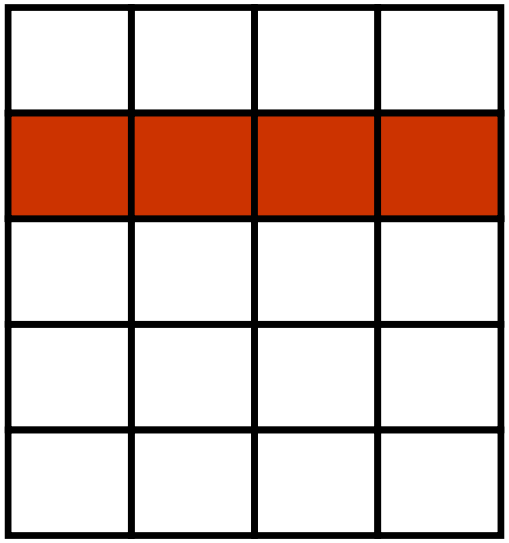


```
for (i=0;i<n;i++){
    calcolare la somma della i-sima riga;
    media_riga[i]=somma/m;
    }
```

media_riga



A



```
for (i=0;i<n;i++)
    {
     somma = 0.0;
     for (j=0;j<m;j++)
         somma = somma+A[i][j];
     media_riga[i] = somma/m;
    }
```

```c
#include <stdio.h>
void visualizza_a2DD(double [][100],int,int);
void visualizza_aI(double [],int);
void main ()
{
 double A[100][100],media_riga[100], somma;
 int i,j,n,m;
 printf("inserire il numero di righe (<=100):");
 scanf("%d",&n);
 printf("inserire il numero di colonne (<=100):");
 scanf("%d",&m);
 for (i=0;i<n;i++)
     for (j=0;j<m;j++)
       scanf("%lf",&A[i][j]);
 for (i=0;i<n;i++)
     {
       somma = 0.0;
       for (j=0;j<m;j++){
                somma = somma+A[i][j];
                }
       media_riga[i] = somma/m;
      }
 printf ("le medie delle %d righe:\n",n);
 visualizza_aI(media_riga,n);
 printf ("l'array 2D e'\n");
 visualizza_a2DD(A,n,m);
}
```

esercizio

realizzare una function C che, dato un array 2D che rappresenta una immagine in bianco/nero, trasformi l'array in modo che esso rappresenti il **negativo** dell'immagine di partenza

una immagine in bianco/nero è rappresentabile con un array 2D di **`short`** in cui ogni elemento rappresenta un pixel e il valore dell'elemento è il livello di grigio del pixel corrispondente

immagine in bianco/nero di al più 256x256 pixel, 256 livelli di grigio (0 →nero, 255 →bianco)

```
/* negativo di una immagine bianco/nero a
   256 livelli di grigio                                    */
void negativo_immagine(short foto[][256],int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
           foto[i][j] = 255 - foto[i][j];
}
```

```c
#include <stdio.h>
void negativo_immagine(short [][256],int,int);
void visualizza_a2DS(short [][256],int,int);
void main()
{
  short mia_foto[256][256];
/* ….qui deve essere definito l'array mia_foto ….*/
  negativo_immagine(mia_foto,128,128); /* mia_foto è solo    */
  visualizza_a2DS (mia_foto,128,128);  /* 128x128 pixel      */
}
/* negativo di una immagine bianco/nero a
   256 livelli di grigio                                  */
void negativo_immagine(short foto[][256],int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
              foto[i][j] = 256 - foto[i][j];
}
void  visualizza_a2DS(short a[][256], int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
   {
     for (j=0;j<m;j++)
         printf("%5d", a[i][j]);
     printf("\n");
   }
}
```

esercizio

realizzare una function C che, dato un array 2D che rappresenta una immagine in bianco/nero, trasforma l'array in modo che esso rappresenti l'immagine di **massimo contrasto** dell'immagine di partenza

una immagine in bianco/nero è rappresentabile con un array 2D di **`short`** in cui ogni elemento rappresenta un pixel e il valore dell'elemento è il livello di grigio del pixel corrispondente

immagine in bianco/nero di al più 256x256 pixel, 256 livelli di grigio (0 →nero, 255 →bianco)

in una immagine in bianco/nero di **massimo contrasto** ogni pixel ha valore bianco (0) oppure nero (255)

```
/* massimo contrasto di una immagine bianco/nero a
   256 livelli di grigio                                */
void max_contrasto_immagine(short foto[][256],
                            int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
          if(foto[i][j] <= 127 && foto[i][j]>=0)
              foto[i][j] = 0;
            else
              foto[i][j] = 255;

}
```

```c
#include <stdio.h>
void max_contrasto_immagine(short [][256],int,int);
void visualizza_a2DS(short [][256],int,int);
void main()
{
  short mia_foto[256][256];
/* ….qui deve essere definito l'array mia_foto ….*/
  max_contrasto_immagine(mia_foto,128,128); /* mia_foto è solo    */
  visualizza_a2DS (mia_foto,128,128);       /* 128x128 pixel      */
}
/* massimo contrasto di una immagine bianco/nero a
   256 livelli di grigio                                  */
void max_contrasto_immagine(short foto[][256],int n,int m)
{
 int i,j;
 for (i=0;i<n;i++)
      for (j=0;j<m;j++)
          if(foto[i][j] <= 127 && foto[i][j]>=0)
               foto[i][j] = 0;
            else
               foto[i][j] = 255;
}
```