

# Matrici come array di puntatori a righe

Una rappresentazione che risolve il problema di  
passare array multidimensionali a funzioni

# Problema :funzione per sommare matrici m per n

```
#include <stdio.h>
#define M 2
#define N 3
```

```
void somma_mat(int a[][M], int b[][M], unsigned n )
{
int i, j;
for (i=0; i<n; i++)
    for (j=0; j<M; j++)
        a[i][j]+= b[i][j]; }
```

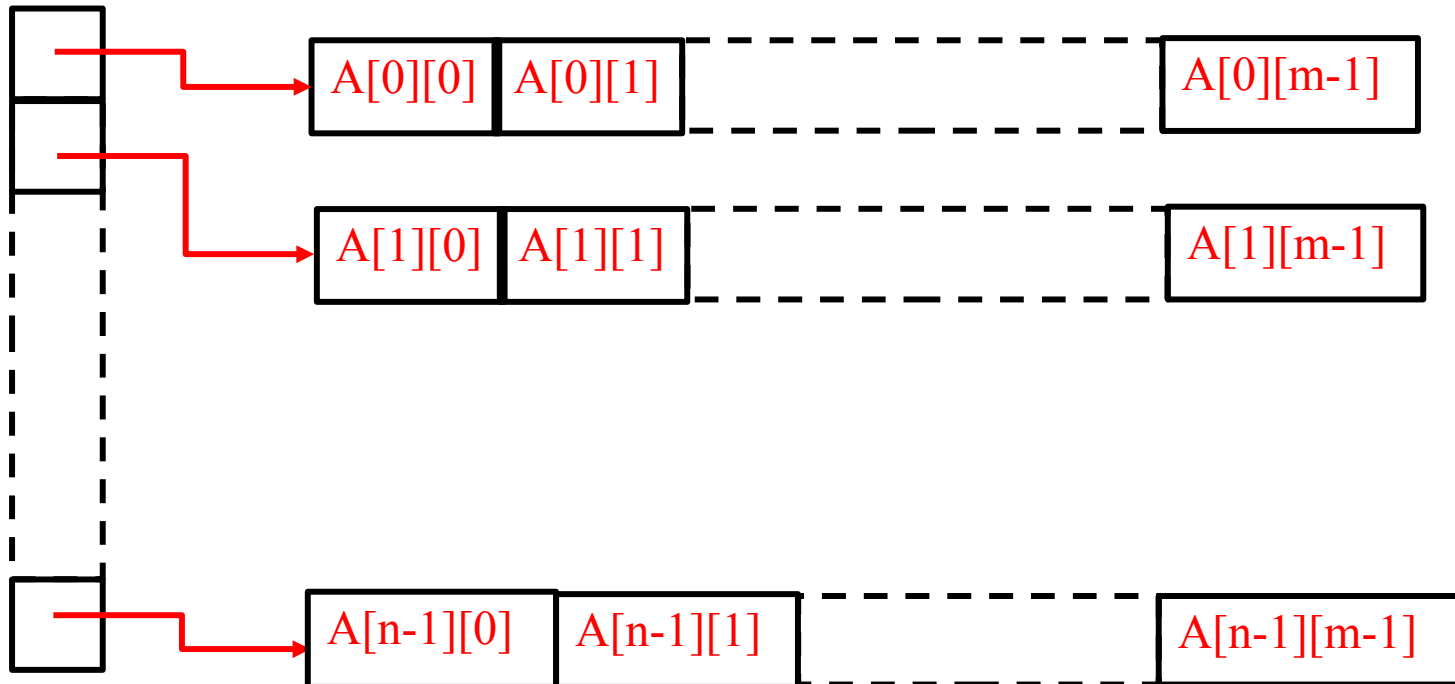
```
void new_mat(int a[][M], unsigned n ) {
int i,j;
for (i=0; i<n; i++)
    for (j=0; j<M;j++)
        scanf("%d",&(a[i][j]));}
```

```
void stampa(int a[][M], unsigned n ) {
```

```
    int i,j;  
    for (i=0; i<n; i++)  
        for (j=0; j<M;j++)  
            printf("%d ", a[i][j]);  
    printf("\n"); } }
```

```
int main (void){  
    int a[N][M];  
    int b[N][M];  
    new_mat(a, N);  
    new_mat(b, N);  
    somma_mat(a, b, N);  
    stampa(a, N);  
}
```

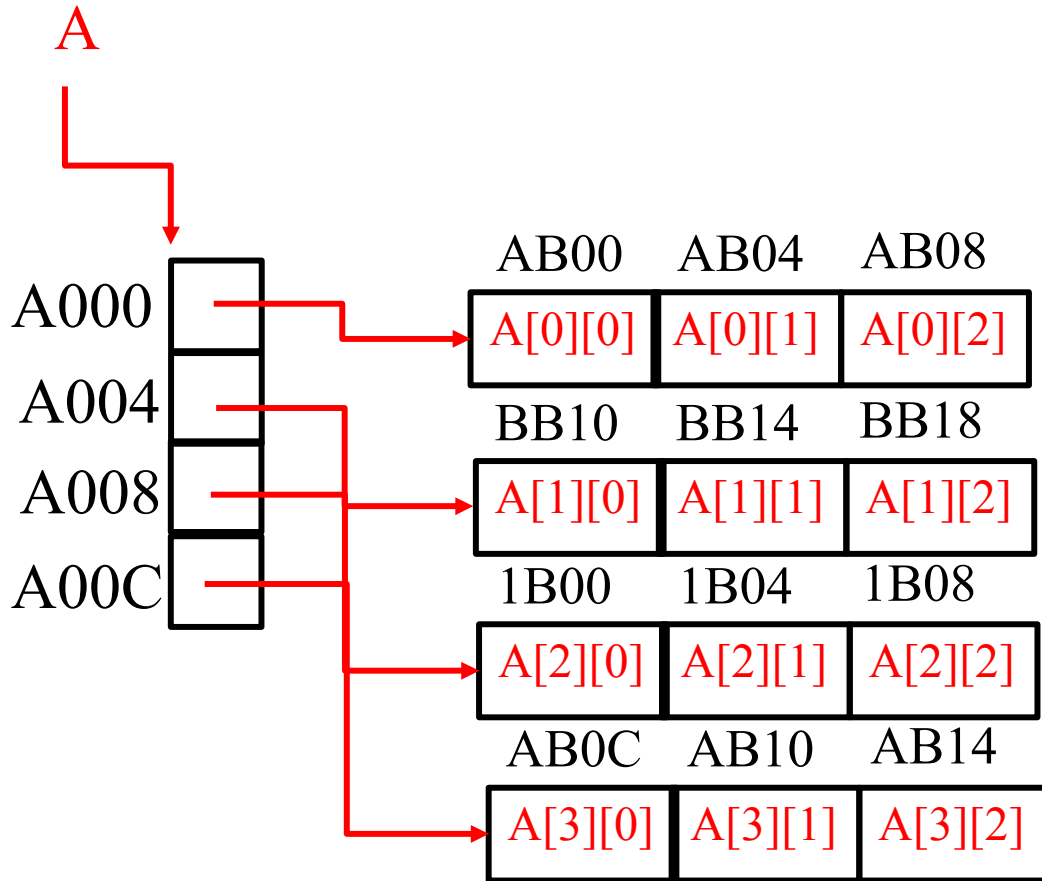
# Esempio: matrice $n$ per $m$



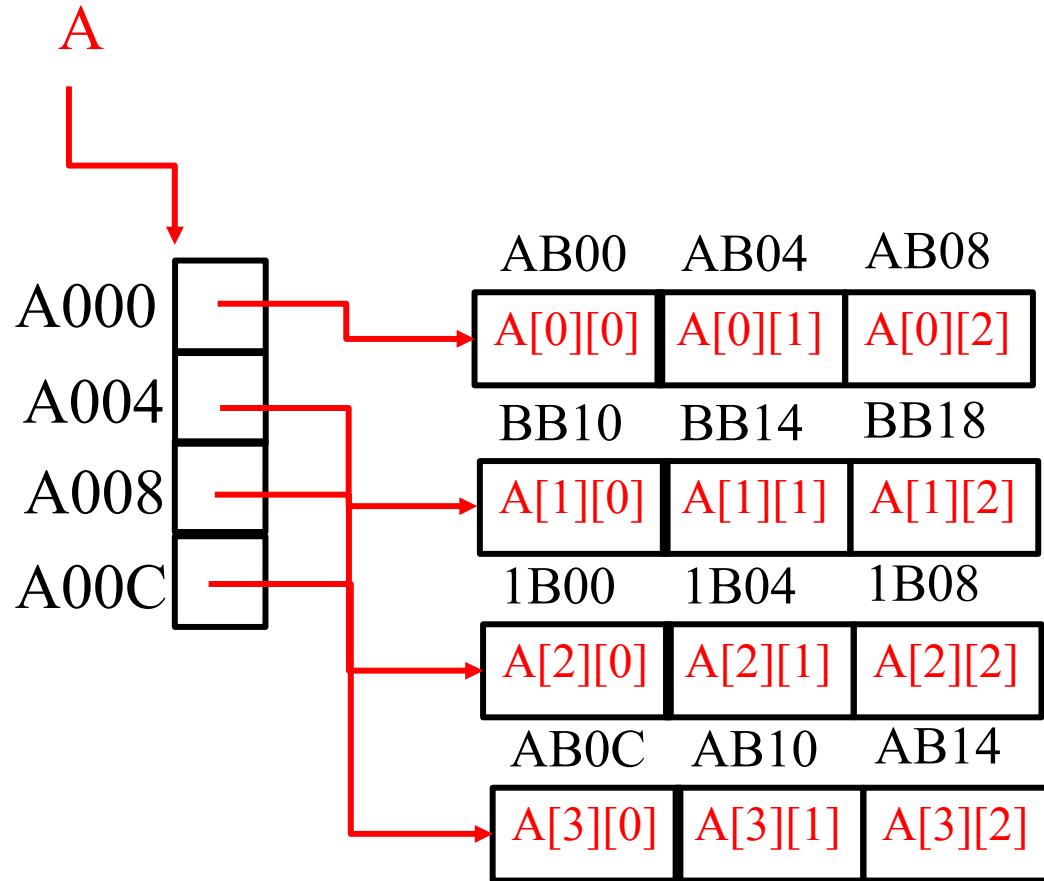
# Cosa vogliamo poter fare

```
/* somma di matrici */  
  
void somma_mat(double** a, double ** b, unsigned n,  
unsigned m ) {  
  
    int i, j;  
  
    for (i=0; i<n; i++)  
        for (j=0; j<m; j++)  
            a[i][j]+= b[i][j];  
  
}
```

# Esempio: matrice 4 per 3



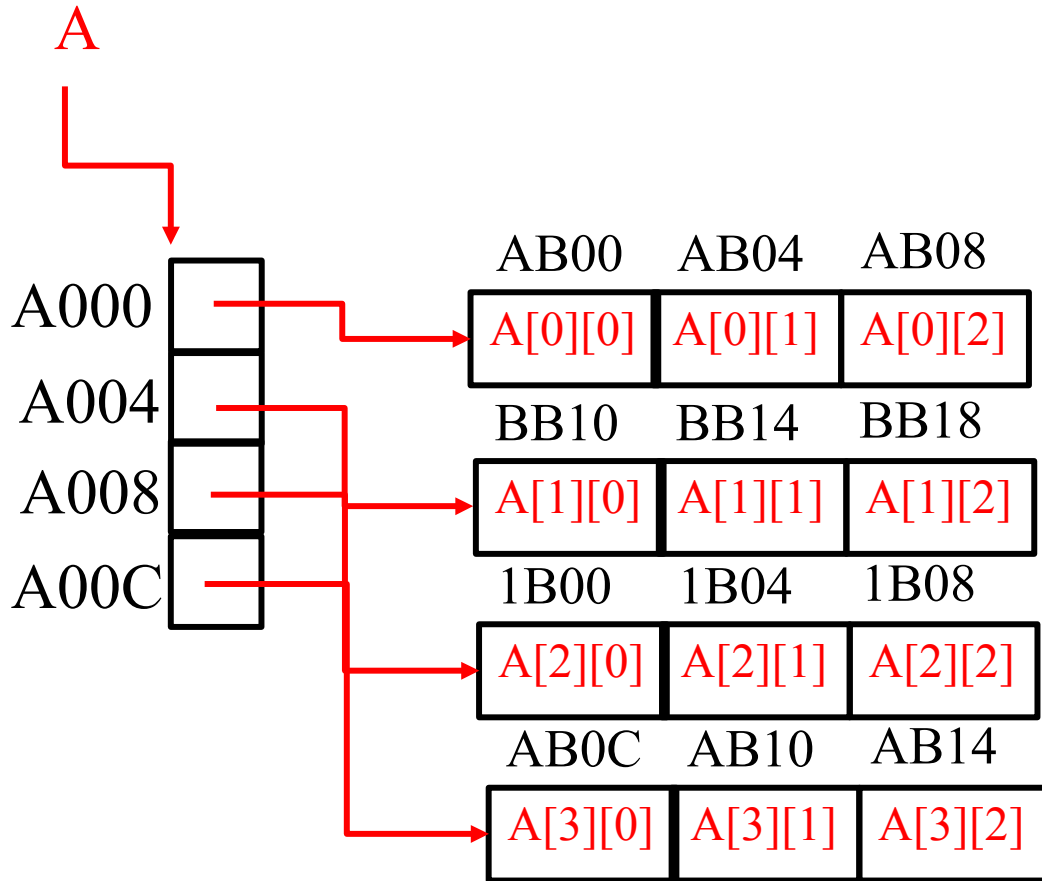
# Esempio: matrice 4 per 3



Il nome della matrice  
È il puntatore A  
double \*\*

L'accesso a un elemento  
avviene correttamente  
con l'usuale operatore  
[...][...]

# Esempio: matrice 4 per 3



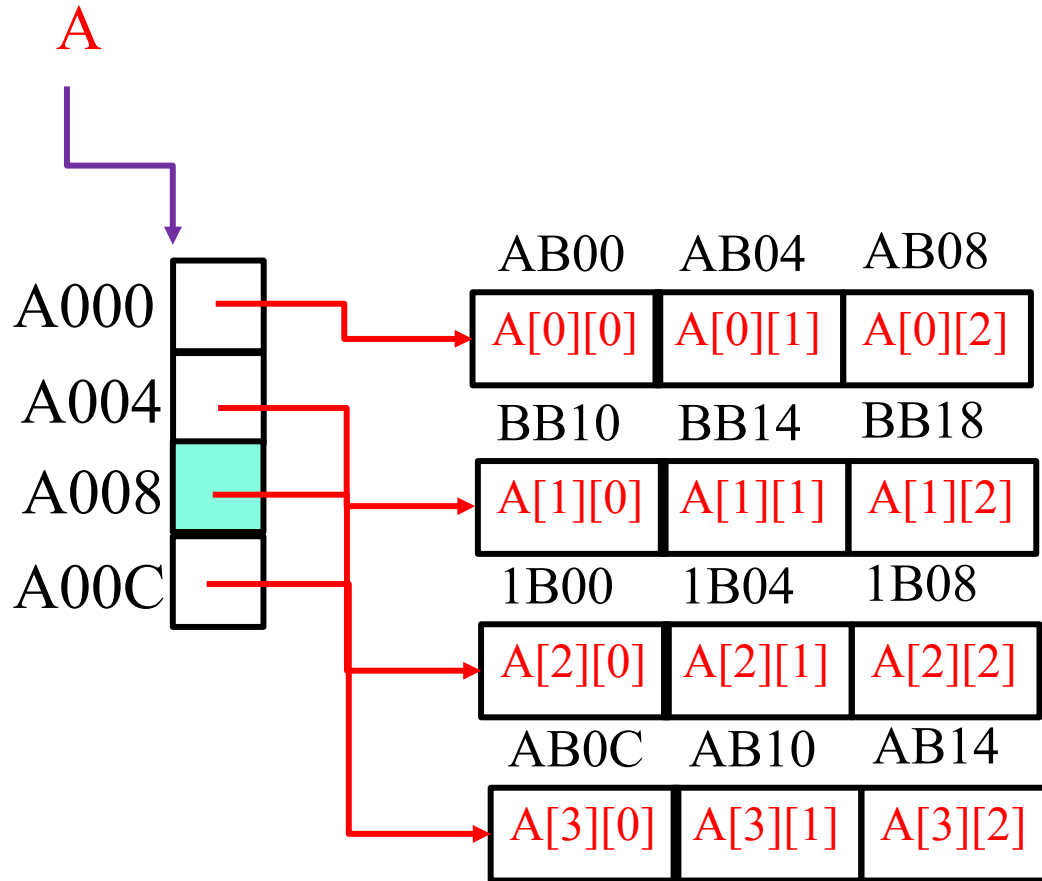
Esempio di accesso

$A[2][2]$

$(A[2])[2]$



# Esempio: matrice 4 per 3



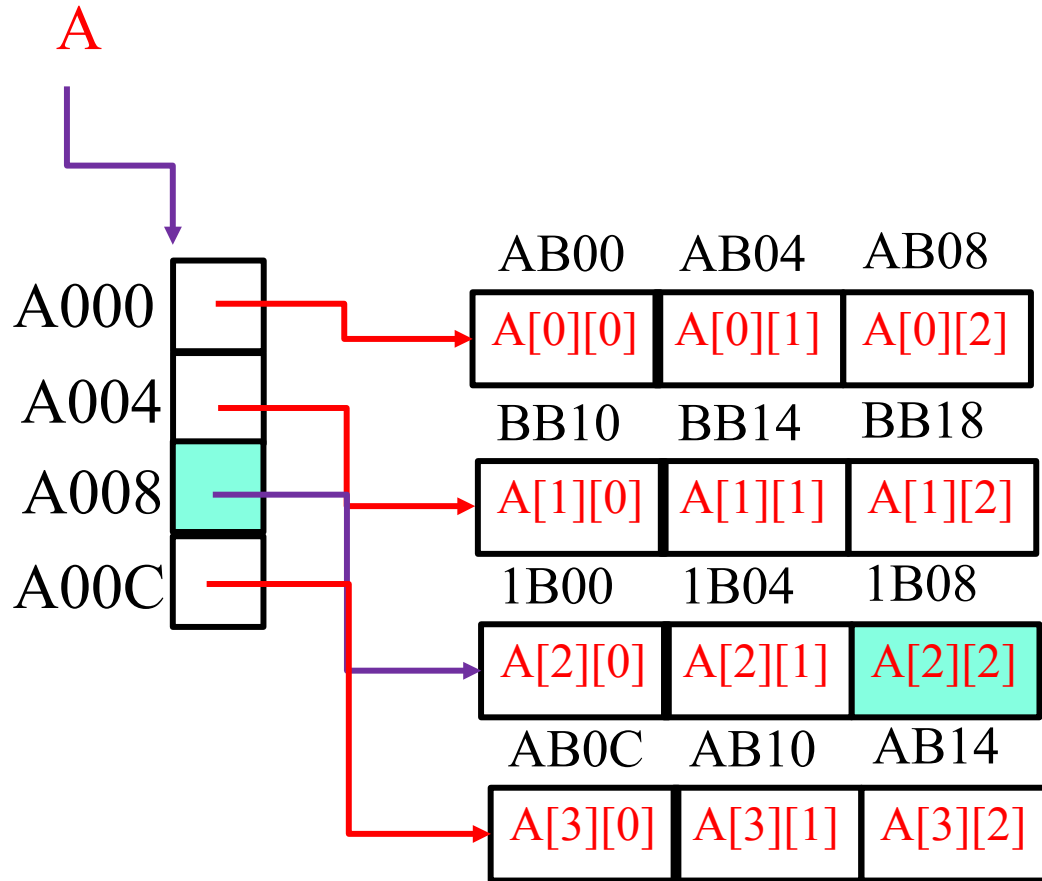
Esempio di accesso

$A[2][2]$

$(A[2])[2]$

$(*(A + 2))[2]$

# Esempio: matrice 4 per 3



Esempio di accesso

$A[2][2]$

$(A[2])[2]$

$*(A + 2)[2]$

$*(1B00 + 2)$

# Esempio: allocazione

```
/* creazione di una matrice (nXm) */  
void new_mat(int ***a, unsigned n, unsigned m ) {  
    int i,j;  
  
    *a=malloc(n*sizeof(int * ));  
    for (i=0; i<n; i++)  
        {(*a)[i]=malloc(m*sizeof(int));  
        for (j=0; j<m;j++)  
            { printf("immetti intero\n");  
              scanf("%d", ((*a)[i])+j);}  
        }  
}
```

# Esempio: deallocazione

```
/* deallocazione di una matrice */  
void free_mat(int*** a, unsigned n ) {  
    int i;  
    for (i=0; i<n; i++)  
        free((*a)[i]);  
    free(*a);  
    *a=NULL;  
}
```

# Esempio: somma di matrici

```
void somma_mat(int** a, int ** b, unsigned n, unsigned m )
{
int i, j;
for (i=0; i<n; i++)
    for (j=0; j<m; j++)
        a[i][j]+= b[i][j]; }
}
```

```
void new_mat(int *** a, unsigned n, unsigned m ) {
int i,j;
```

```
*a=malloc(n*sizeof(int * ));
for (i=0; i<n; i++)
    {(*a)[i]=malloc(m*sizeof(int));
    for (j=0; j<m;j++)
        { printf("immetti intero\n");
        scanf("%d", ((*a)[i])+j);}}
```

```
}
```

```
void stampa(int ** a, unsigned n, unsigned m ) {  
    int i,j;  
    for (i=0; i<n; i++)  
        {for (j=0; j<m;j++)  
            printf("%d ",a[i][j]);  
            printf("\n");}}
```

```
void free_mat(int*** a, unsigned n ) {  
    int i;  
    for (i=0; i<n; i++)  
        free((*a)[i]);  
    free(*a);  
    *a=NULL;  
}
```

```
int main (void){  
    int ** a;  
    int ** b;  
    new_mat(&a, N, M);  
    new_mat(&b, N, M);  
    somma_mat(a, b, N, M);  
    stampa(a, N, M);
```