



Passaggio di parametri di tipo array

- Per gli array monodimensionali non va specificata la dimensione tra le parentesi, sia nel prototipo che nell'intestazione:

```
void invertiArray(int [], int);
```

- È necessario passare come parametro la dimensione dell'array
- Per le **stringhe** non serve passare la dimensione come parametro poiché il contenuto valido termina con il carattere `'\0'`
- Non si può specificare un array con un valore di return: *verrebbe restituito un indirizzo, non ha senso*
- Per gli **Array multidimensionali** si deve specificare tutte le dimensioni dalla seconda in poi (*per permettere la delinearizzazione della matrice*), sia nel prototipo che nell'intestazione

```
void trasponiMatrice (int a[][N], int dim1);
```

- Si ricordi che il nome della matrice è un puntatore a vettore costante, quindi il seguente prototipo è equivalente:

```
void trasponiMatrice (int (*a)[N], int dim1);
```

- Nell'invocazione specificheremo solo il nome della matrice:

```
int b[M][N];
```

```
....
```

```
trasponiMatrice(b, M);
```

- Per ovviare a questo "problema"/"limite" si può passare l'indirizzo della prima cella e gestire esplicitamente la linearizzazione della matrice

```
void trasponiMatrice (int *a, int dim1, int dim2);
```

- Nell'invocazione specificheremo l'indirizzo della prima cella

```
int b[M][N];
```

```
....
```

```
trasponiMatrice (&b[0][0], N, M);
```