Riassumendo: array e puntatori

- Con la seguente dichiarazione:

```
int a[5], i, * p;
```

`a[i]`	equivale a `*(a + i)`
`p = a`	equivale a `p = &a[0];`
`p = a + 1`	equivale a `p = &a[1];`
`a = p;`	è un ERRORE !!
`a = a + 1;`	è un ERRORE !!

- **Cioè occorre ricordare che `a` è un array !!**

Ancora sull'aritmetica dei puntatori

- Se `p` e `q` puntano a due diversi elementi di uno stesso array, la differenza:

  `p - q`

  dà la distanza **nell'array** tra gli elementi puntati
- Tipicamente **non** coincide con la differenza "aritmetica" tra i valori numerici dei due puntatori
  - È una distanza espressa in "numero di elementi"

Array multidimensionali

Per gli array multi-dimensionali

- Il calcolo dello **spiazzamento** richiede di conoscere le dimensioni intermedie
  - **`Tipo m[R][C];`** /*N.B.: R righe, C colonne*/
  - **`m[i][j]`** → accesso al j-esimo elemento della i-esima colonna
  - **`m[i][j]`** ≡ **`*( *(m + i) + j )`** ≈ m + **C**·i + j
    - serve conoscere sia la dimensione del tipo sia il numero di colonne (sizeof(Tipo) e C; *la "altezza" R non serve*)
  - Tipo p[**X**][**Y**][**Z**]
  - p[i][j][k] ≡ *(*(*(p+i)+j)+k) ≈ p + **Y**·**Z**·i + **Z**·j + k
    - serve conoscere dimensione del tipo, altezza e larghezza (sizeof(Tipo), Y e Z; *la "profondità" X non serve*)