

## QUALIFICATORI:

IL TIPO CON UN QUALIFICATORE VIENE CHIAMATO **TIPO QUALIFICATO**;

# CONST, VOLATILE, RESTRICT!

SERVONO A DARE INFORMAZIONI AGGIUNTIVE AL COMPILATORE.

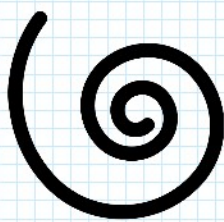
- **const** (C89): indica che i dati sono in sola lettura.

- **volatile** (C89): indica che un valore può cambiare tra due accessi diversi, anche se apparentemente nulla l'ha cambiato (previene ottimizzazioni del compilatore).

*È una variabile che spesso modifica, quindi accessi in memoria spesso per rileggerla;*

- **restrict** (C99): si utilizza solo con dichiarazioni di puntatori. Indica al compilatore che, per tutta la vita del puntatore, solo questo avrà accesso all'oggetto puntato.

- I qualificatori di tipo possono essere utilizzati per differenziare i puntatori e l'area di memoria puntata



## REGOLA DELLA SPIRALE!

- **int** \*ptr: un puntatore a intero
- **int const** \*ptr: un puntatore a **intero costante**
- **const int** \*ptr: un puntatore a intero costante
- **int \*const** ptr: un **puntatore costante** a intero

→ IL DATO PUNTATO CAMBIA, IL PUNTATORE NO!

IL DATO PUNTATO NON CAMBIA!  
IL PUNTATORE SÌ.

- **const int \*const** ptr: un puntatore costante a intero costante
- **int \*\*ptr**: un puntatore a un puntatore a intero
- **int \*\*const** ptr: un puntatore costante a un puntatore a intero
- **int \*const \*** ptr: un puntatore a un puntatore costante a intero

- **char \*\*argv**: un puntatore a puntatore a carattere
- **const char \*const \*const** ptr: un puntatore costante a un puntatore costante a un carattere costante
- **volatile char \*const** ptr: un puntatore costante a un carattere volatile

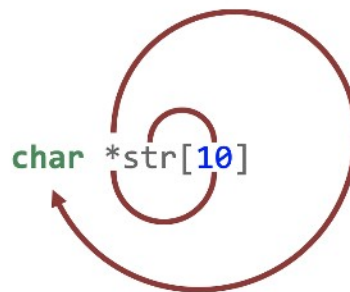


- `int *ptr[13]`: un vettore di 13 puntatori a intero
- `int (*ptr)[13]`: un puntatore a un vettore di 13 interi
- `void *ptr()`: una funzione che restituisce un puntatore generico
- `void (*ptr)()`: un puntatore a una funzione senza valore di ritorno
- `char ((*x())[5])()`: una funzione che restituisce un puntatore a un vettore di puntatori a funzione che restituiscono un carattere
- `char ((*x[3])())[5]`: un vettore di tre puntatori a funzione che restituiscono un puntatore ad un vettore di cinque caratteri

## APPLICHIAMO la Regola della spirale!

- La regola della spirale è una “regola empirica” che permette di decifrare qualsiasi tipo C
- La “lettura” di un tipo parte dal nome della variabile e segue una spirale in senso orario
- Ci si ferma quando si sono presi in considerazione tutti gli elementi
- Ciò che si trova fra parentesi ha la precedenza!

1



`str` è un vettore di 10 puntatori a carattere



2

`char *(*fp)(int, float *)`



fp è un puntatore a una funzione che accetta un intero e un puntatore a float che restituisce un puntatore a carattere

3

`void (*signal(int, void (*fp)(int)))(int)`



signal è una funzione che accetta un intero e un puntatore a una funzione che accetta un intero e non restituisce nulla che restituisce un puntatore a una funzione che accetta un intero e non restituisce nulla

4

`char ((*x[3])())[5]`

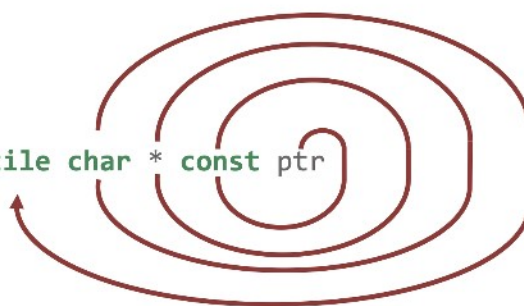


x è un vettore di tre puntatori a funzione che restituiscono un puntatore ad un vettore di cinque caratteri



5

`volatile char * const ptr`



ptr è un puntatore costante a un carattere volatile