

# Espressioni e strutture di controllo

Matteo Spanio

19 marzo 2024

Il C è basato su *espressioni*. Queste espressioni sono un insieme di **variabili**, **costanti** e **operatori**.

## Operatori

Ogni operatore ha varie caratteristiche:

- **arietà**: numero di argomenti (unario, binario, ternario)
- **precedenza**: ordine in cui vengono analizzati gli operatori
- **associatività**: se ci sono più operatori uguali (o di uguale precedenza)
- **side-effects**, se presenti (cioè le variabili coinvolte vengono cambiate o no)

## Espressioni istruzione

Tutte le espressioni possono essere anche *statement*: in una linea posso avere anche solo una operazione singola seguita da ;

```
i = 1;      // utile
i++;       // utile
i * j + 2   // inutile, potrebbe dare warning
           // "statement with no effect"
```

Nel secondo caso il risultato viene scartato, ma la modifica avviene lo stesso.

## Espressioni logiche

Alcuni operatori ci dicono se una certa espressione è “vera” o “falsa”, ad esempio:

`i < j && j < k` ci dice se *i* è minore sia di *j* che di *k*

**i** Nota

$i < j < k$  si può scrivere ma non fa quello che sembra, in realtà:  $(i < j) < k$  è vero con `int i=3, j=2, k=1`.

Il tipo di ritorno di queste espressioni in C è un intero, il tipo booleano non esiste in C (in C99 si è provato a sistemare).

- $0 \rightarrow$  falso
- $1 \rightarrow$  vero (e tutti gli interi diversi da 0)