

Operatori di incremento e decremento

++ e **--** sono operatori unari con la stessa priorità a del meno unario e associatività da destra a sinistra.

Si possono applicare solo a variabili (di tipi interi, floating o puntatori), ma non a espressioni generiche (anche se di questi tipi).

Esempio: ipotizziamo di applicare, dopo la dichiarazione e la relativa inizializzazione della variabile *c*, una istruzione alla volta tra quelle seguenti:

```
int c = 5;           /* dichiarazione variabile di esempio */
1a) printf("%d", c++); /* visualizzato il valore 5 ma dopo c varrà 6 */
1b) printf("%d", c--); /* visualizzato il valore 5 ma dopo c varrà 4 */
1c) printf("%d", --c); /* visualizzato il valore 4 e c dopo varrà esattamente 4 */
1d) printf("%d", ++c); /* visualizzato il valore 6 e c dopo varrà esattamente 6 */
```

Semantica degli operatori di incremento/decremento

Postfisso: la variabile viene modificata PRIMA di utilizzarne il relativo valore.

Prefisso: la variabile viene modificata DOPO averne utilizzato il relativo valore.

Operatore	Simbolo	Formato	Operazione
incremento postfisso	++	c++	rende disponibile il valore di c, poi lo incrementa
decremento postfisso	—	c—	rende disponibile il valore di c, poi lo decrementa
incremento prefisso	++	++c	incrementa il valore di c, poi lo rende disponibile
decremento prefisso	—	—c	decrementa il valore di c, poi lo rende disponibile

Esempio:

```
int c = 5;
int b = 30 / c++; /* b == 6, c == 6 */
int d = 6 + --c; /* d = 11, c == 5 */
```

In sintesi gli operatori postfissi modificano il valore della variabile dopo l'utilizzo del valore (vecchio) nell'espressione.

Gli operatori prefissi modificano il valore della variabile cui sono applicati prima che se ne utilizzi il valore.

N.B. Ovviamente tali operatori possono comparire in qualunque tipo di espressioni, eventualmente presenti anche all'interno di condizioni logiche, dove vanno opportunamente gestiti.

Esempio 1: cosa verrà mostrato a video?

```
int c = 5;
if (c++ == 5)
{
    printf("ramo allora e c vale %d", c);
}
else
{
    printf("ramo altrimenti e c vale %d", c);
}
```

A

```
int c = 5;
if (++c == 5)
{
    printf("ramo allora e c vale %d", c);
}
else
{
    printf("ramo altrimenti e c vale %d", c);
}
```

B

Soluzione 1:

caso **A**) ramo "else" e visualizzazione del valore 6

caso **B**) ramo "if" e visualizzazione valore 6

Esempio 2: cosa verrà mostrato a video?

```
int c = 4;
if (c-- == 4)
{
    printf("ramo allora e c vale %d", c);
}
else
{
    printf("ramo altrimenti e c vale %d", c);
}
```

A

```
int c = 4;
if (--c == 4)
{
    printf("ramo allora e c vale %d", c);
}
else
{
    printf("ramo altrimenti e c vale %d", c);
}
```

B

Soluzione 2:

caso **A**) ramo "else" e visualizzazione del valore 3

caso **B**) ramo "if" e visualizzazione valore 3