

Soluzione: bit di complemento per rappresentare numeri negativi

Davide Gaetano

Novembre 2025

1 Introduzione sul bit di complemento

In C un tipo "signed" può rappresentare sia numeri positivi sia numeri negativi. Sappiamo che i numeri positivi e quelli negativi hanno lo stesso numero di bit; ciò significa che metà del range dei numeri positivi deve essere utilizzata per rappresentare anche quelli negativi.

2 Spiegazione

2.1 Unsigned char

Per esempio, se prendiamo un "unsigned char" (ovvero un numero senza segno), sappiamo che può rappresentare una cifra da 0 a 255.

In binario **00000000 a 11111111** e in esadecimale **0x00 a 0xFF**.

2.2 Signed char

Se invece prendiamo un "signed char" (ovvero un numero con segno), può rappresentare numeri da -128 a -1 e da 0 a 127.

Per i numeri negativi, in binario **10000000 a 11111111** e in esadecimale **0x80 a 0xFF**.

Per i numeri positivi, in binario **00000000 a 01111111** e in esadecimale **0x01 a 0x7F**.

Notiamo quindi che il bit più a sinistra è quello che determina se il numero è positivo o negativo.

1/2 points (ungraded)

Figure 1: Esercizio

3 Risoluzione dell'esercizio

L'esercizio richiesto da Xeno era il seguente:

Qual è la rappresentazione in esadecimale, a 2 complementi, del valore più basso possibile in un valore a 8 byte signed?

- Sappiamo che 8 byte sono 64 bit, quindi un bit verrà utilizzato per rappresentare la negatività del numero e il resto saranno zeri (poiché è richiesto di rappresentare il numero più basso).
- Quindi 63 zeri e un uno equivalgono a:
1000
- Che in esadecimale corrisponde a: **8000000000000000**.
- Che in decimale corrisponde a: **-9223372036854775808**.
- Esercizio completato.