

VERIFICA DI INFORMATICA (SOLUZIONE)

Liceo Scientifico "A. Volta"
classe 3° B, 22/12/2007
prof. Magni Claudio

1)

$$10110_2 = 1 * 2^4 + 1 * 2^2 + 1 * 2^1 = 16 + 4 + 2 = 22_{10}$$

2)

34	0	↑	$34_{10} = 100010_2$
17	1		
8	0		
4	0		
2	0		
1	1		
0			

3)

175	15 = F	↑	$175_{10} = AF_{16}$
10	10 = A		
0	0		

4)

$$4E_{16} = 4 * 16^1 + 14 * 16^0 = 64 + 14 = 78_{10}$$

5)

Isolo i bit a gruppi di 4 e converto in una cifra esadecimale: $\underbrace{10}_2 \underbrace{1110}_{14} \rightarrow = 2E_{16}$

6)

$$10110_2 = -1 * 2^4 + 1 * 2^2 + 1 * 2^1 = -16 + 4 + 2 = -10_{10}$$

7)

Prima trovo +28:

28	0	↑
14	0	
7	1	
3	1	
1	1	
0		

$$28_{10} = 11100_2 \rightarrow +28_{10} = 011100_2$$

Ora complemento i bit e sommo 1:

$$100011 + 1 = 100100$$

$$-28_{10} = 100100_2$$

8)

$$6_{10} = 110_2$$

$$8_{10} = 1000_2$$

Eseguo la somma:

$$0110 + 1000 = 1110_2 = 14_{10}$$

9)

Entrambe le codifiche esprimono numeri interi.

In modulo e segno il primo bit (quello più a sx) è utilizzato per indicare il segno: 0 = +, 1 = -.

Il complemento a due è stato inventato per permettere al processore di fare facilmente le somme algebriche: infatti il modulo e segno utilizza un bit di segno per mancanza di altri simboli nel calcolatore, ma questo porta a dei numeri negativi scritti in modo "errato". Nel complemento a due, pur mantenendo il primo bit di segno come il modulo e segno, la codifica dei numeri negativi è diversa dalla semplice anteposizione del bit 1: è più complessa, ma permette al processore di fare la somma algebrica semplicemente sommando i numeri.

10)

Byte = 8 bit

Byte si abbrevia in 'B', mentre bit in 'b'.

Kilo e Mega sono prefissi che moltiplicano l'unità di misura a cui si riferiscono per, rispettivamente, 2^{10} e 2^{20} .

$$1 \text{ KB} = 1 \text{ Kilobyte} = 2^{10} \text{ B} = 1024 \text{ B} = 8 * 1024 \text{ bit}$$