Esercizi di preparazione al primo test: conversioni e codici

Esercizio 1. Convertire i seguenti numeri da decimale a esadecimale (base 16) e poi da esadecimale a binario

a) 757 b) 123

Soluzione:
Si usa il metodo delle divisioni ripetute per la base di b arrivo (16) nella matematica della base a di partenza (10) con accumulo dei resti **da destra verso sinistra**, e loro trasformazione nella base b di arrivo (16).

16 | 757
16 | 47 r5
16 | 2 r15=F_{16}
0 r2

in decimale, i resti sono: 5, 15, e 2. ma i resti vanno poi rappresentati in esadecimale, e si ha:
5_{10}->5_{16} 15_{10} ->F_{16} 2_{10}->2_{16} quindi 757_{10}=$2F5_{16}$

Ora convertiamo da esadecimale a binario. Questo è un caso di conversione da una base 2^i e base due. In questo caso basta associare ad ogni simbolo della base di partenza i bit della base due.

2 -> 0010
F-> 1111
5-> 0101

0010 1111 0101
2 F 5

Quindi il risultato è 00101110101

b) 123

16 | 123
16 | 7 r11
0 r7

11_{10}=B_{16} e dunque

123_{10}= $7B_{16}$

Analogamente all'esercizio a), ottenete : $B7_{16}$ =01111011_2

0111 1011
7 B

Esercizio 2. Convertire:
a) 0,625 da base 10 a binario accumulando al massimo 8 cifre
b) 0,25 e 0,17 da base 10 a esadecimale, accumulando al massimo 3 cifre

Soluzione:
Si usa il metodo delle moltiplicazioni ripetute della parte decimale del numero per la base di arrivo b. le moltiplicazioni sono eseguite nella matematica della base di partenza a, con “accumulo” delle parti intere del risultato della moltiplicazione, **da sinistra verso destra**.

Per l’esercizio a) si ha:

0,625x2=1,4 i1=1
0,4x2=0,8 i2=0
0,8x2=1,6 i3=1
0,6x2=1,2 i4=1
0,2x2=0,4 i5=0

Siccome 0,4 è stato ottenuto in precedenza (al passo 2), il processo si ripete e si continua ad ottenere sempre la stessa sequenza di 4 bit.
Quindi il risultato è 0,10110 0110 0110 ....
Poichè si chiede di arrestarsi dopo 8 iterazioni, il risultato è 0,10110011

per l’esercizio b) si ha:

0,25x16=4,00
poichè la parte decimale è zero, ci si arresta alla prima iterazione.
Dunque $0{,}25_{10}=0{,}4_{16}$

Per il secondo numero si ha:

$$\begin{array}{r} 0.17 \\ \underline{16} \\ (2).72 \\ \underline{16} \\ (11).52 \\ \underline{16} \\ (8).32 \end{array}$$

Le parti intere sono 2, 11 e 8 (espresse in base 10). 2 e 8 hanno la stessa rappresentazione in base 16, mentre $11_{10}=B_{16}$
Quindi $0{,}17_{10}= 0{,}2B8_{16}$

Esercizio 3. Convertire $3BA.25_{14}$ in base 10

Soluzione: usare il metodo polinomiale

$$\begin{aligned} 3BA.25_{14} &= 3 \times 14^2 + 11 \times 14^1 + 10 \times 14^0 + 2 \times 14^{-1} \\ &\quad + 5 \times 14^{-2} \\ &= 588 + 154 + 10 + 0.1684 = 752.1684_{10} \end{aligned}$$

Esercizio 4. Sommare e sottrarre a) 1111e 1010, b) 1111 e 1001

Soluzione:

```
  111
  1111   (Add)      1111   (Sub)
 +1010             -1010
 11001              0101
```

```
 111            111
 1111   (Add)   1111   (Subtract)
 1001           1001
11000           0110
```

Esercizio 5 . Eseguite (-11) + (-4) in complemento a due. Di quanti bit avete bisogno? Ricalcolate poi il risultato in decimale.

Soluzione:

1) di quanti bit avete bisogno? In generale, come visto a lezione, con n cifre e base b potete rappresentare
dal valore b^{n-1} -1 al valore $-b^{n-1}$. Ponendob=2 e n=4, ottenete il range [+7,-8] che non vi basta, dovendo rappresentare il numero -11. Con n=5 ottenete il range [+15, -16] che invece è sufficente. Perciò vi servono minimo 5 bit. Volendo rappresentare, come è più logico, anche il valore +16, è conveniente usare 6 bit.

2) per calcolare -N in Ca2 tenete presente che $-N = \overline{N} + 1$, quindi:
   a) convertite 11 in binario e con 5 bit: 001011
   b) complementate: 110100
   c) sommate 1: 110101
   d) analogamente, -4= 111100

   quindi:

```
   110101
   111100
(1)110001
```

Notate che c'è un "overflow" di un bit. In generale, la somma di due numeri a k bit richiede k+1 bit.

3) Convertiamo ora in decimale usando la formula:

$$-c_{n-1}b^{n-1} + \sum_{0}^{n-2} c_i b^i$$

Applicando al nostro caso, abbiamo b=2 e n-1=6, e dunque:

$$-1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 =$$
$$-64 + 32 + 16 + 1 = -15$$

Esercizio 6. Sia:

$$\begin{aligned} s &= 1 \\ e &= 10000111 \\ m &= 11011000000000000000000 \end{aligned}$$

un numero in virgola mobile secondo lo standard IEEE 754 a 32 bit. Convertire in decimale.

Soluzione:

La formula generale è $(-1)^s \cdot 2^{(e-127)} \cdot 1{,}m$ dove s, e ed m vanno espressi in base 10.

Abbiamo
$e = 10000111_2 = 135_{10}$ (usando il metodo polinomiale)

e quindi:

$$\begin{aligned} N &= (-1)^s \cdot 2^{(e-127)} \cdot 1.m \\ &= -1 \cdot 2^{135-127} \cdot 1.11011 \\ &= -1 \cdot 2^8 \cdot 1.11011 \\ &= -111011000_2 \\ &= -(2^8 + 2^7 + 2^6 + 2^4 + 2^3)_{10} \\ &= -472_{10} \end{aligned}$$

Esercizio 7
Convertire 11,876 in rappresentazione a virgola mobile con 32 bit (1 per i segno, 8 per l'esponente ed il resto per la mantissa normalizzata), e esponente in complemento a due.

Soluzione:
Passo 1: trasformazione da base a *a* base *b* in virgola fissa
Usando il metodo delle divisioni per 2 ripetute, si ha per la parte intera:
11_{10} --> $\mathbf{1011_2}$

Usando il metodo delle moltiplicazioni ripetute per due, si ha per la parte decimale:
0.876_{10} -> 0.1110000001000001100_2

Passo due: normalizzazione
La mantissa non normalizzata è perciò 1011.1110000001000001100

per normalizzare, occorre spostare la virgola di 4 posizioni a sinistra
$0{,}10111110000001000001100 * 2^4$

Passo 3: calcolo dell'esponente e conversione in base b
Quindi l'esponente è 4, che in binario è 0100 ed in complemento a due con 8 bit è 00000100.

<u>In conclusione</u>:

S	esponente	mantissa
0	00000100	10111110000001000001100
31	30 23	22 0