

Corso di

Fondamenti di Informatica 2

per il corso di Laurea di Ingegneria Gestionale

Esercizi sulle codifiche numeriche e di testo con soluzioni

Università degli Studi di Udine - A.A. 2010-2011

Docente Ing. Sandro Di Giusto Ph.D.

- Si convertano i seguenti numeri dalla base decimale al binario puro:
 - $13_{10} = 1101_2$
 - $137_{10} = 10001001_2$
 - $1039_{10} = 10000001111_2$
 - $-171_{10} = \text{NON E' POSSIBILE}$ (in binario puro si rappresentano solo numeri positivi!)
- Si convertano i seguenti numeri dal binario puro al decimale:
 - $10110_2 = 22_{10}$
 - $1100111_2 = 103_{10}$
 - $001101011_2 = 107_{10}$
 - $1010110101_2 = 693_{10}$

Numeri relativi in binario



- Si convertano i seguenti numeri dalla base decimale a tutte e tre le codifiche viste per i numeri relativi (modulo/segno, complemento a 1 e complemento a 2), indicando anche il numero di bit minimo indispensabile per la rappresentazione. Si scriva infine il numero in complemento a 2 espresso su 16 bit (estensione del segno)

– 27_{10}

– 99_{10}

– -99_{10}

– -569_{10}

– -1025_{10}

Numero dec.	Binario puro	Modulo e segno	Compl. a 1	Compl. a 2	Profond. min in bit	C2 su 16 bit
27_{10}	11011	011011	011011	011011	5	0000000000011011
99_{10}	1100011	01100011	01100011	01100011	7	000000001100011
-99_{10}	Imposs.	11100011	10011100	10011101	7+1=8	1111111110011101
-569_{10}	Imposs.	11000111001	10111000110	10111000111	10+1=11	1111101111000111
-1025_{10}	Imposs.	11000000001	101111111110	101111111111	11+1=12	1111101111111111

Conversioni di base



- Si convertano i seguenti numeri positivi nelle 3 basi mancanti:

- 18_{10}
- 93_{10}
- 210_{10}
- 52_8
- 176_8
- 391_8
- $1D_{16}$
- 49_{16}
- $E07_{16}$
- 101011_2
- 11100011_2
- 11001010111_2

Decimale	Binario puro	Ottale	Esadecimale
18	10010	22	12
93	1011101	135	5D
210	11010010	322	D2
42	101010	52	2A
126	1111110	176	7E
Imposs.	Imposs.	391	Imposs.
29	11101	35	1D
73	1001001	111	49
3591	111000000111	7007	E07
43	101011	53	2B
227	11100011	343	E3
1623	11001010111	3127	657

- Dato un elaboratore operante su 8 bit in complemento a 2, si simulino le seguenti operazioni binarie, convalidando il risultato con l'analisi dei carry ed eseguendo infine la verifica decimale:
 - $28_{16} + 45_{16} = 6D_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $40 + 69 = 109$)
 - $91_{16} + 80_{16} = 11_{16}$ errato $c_n \neq c_{n-1}$ (verifica: $-111 - 128 \neq 17$)
 - $AB_{16} - BA_{16} = F1_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $-85 - (-70) = -15$)
 - $-C0_{16} + 17_{16} = 57_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $-(-48) + 23 = 71$)
- Dato un elaboratore operante su 16 bit in complemento a 2, si simulino le seguenti operazioni binarie, convalidando il risultato con l'analisi dei carry ed eseguendo infine la verifica decimale:
 - $0123_{16} + 6543_{16} = 6666_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $291 + 25923 = 26214$)
 - $B101_{16} - B100_{16} = 0001_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $-20223 - (-20224) = 1$)
 - $ABCD_{16} - 1357_{16} = 9876_{16}$ corretto ovvero $c_n = c_{n-1}$ (verifica: $-21555 - 4951 = -26506$)

- Si scrivano i seguenti numeri decimali in virgola fissa (miglior approssimazione), su 8 bit complessivi di cui 4 per parte intera e 4 per quella frazionaria (e si scriva a fianco il valore realmente codificato):
 - $6.1875_{10} = 01100011$ (pari esattamente a 6.1875_{10})
 - $-2.359375_{10} = 11100101$ (pari a -2.3125_{10})
 - $13.25_{10} =$ non è possibile (parte intera non rappresentabile su 4 bit)
- Si scrivano i seguenti numeri decimali in virgola mobile singola precisione (32 bit) e codificati in esadecimale (8 cifre):
 - $26.024_{10} = 41D1EB85$
 - $-0.0001976_{10} = B94F32D8$
 - $897.5_{10} * 10^5 = 4E55FB0D$ (NOTA: Fattibile agevolmente solo con calcolatore!!!)
- Si scrivano in decimale i seguenti numeri codificati in esadecimale (8 cifre) e rappresentanti numeri in virgola fissa singola precisione:
 - $BD480000_{16} = -0.048828125$

- Si decodifichi il testo associato alla seguente sequenza esadecimale (si usi l'Extended ASCII come tabelle di decodifica e la terminazione di riga Windows ovvero CR+LF)

– 443A 2051 7561 6E74 6F20 6661 2028 3132 2A33 2B38
292F 3131 3F20 0D0A 523A 2046 6120 3421

D: Quanto fa $(12*3+8)/11$? **<CR><LF>**
R: Fa 4!

- Relativamente all'esercizio precedente, si risponda:
 - Il testo ottenuto risulterebbe identico se usassi come tabella di decodifica l'ISO 8859-1? **Sì perché fa uso solo dei primi 128 simboli che sono gli stessi**
 - Il testo ottenuto risulterebbe identico se usassi come tabella di decodifica l'UTF-8? **Sì perché fa uso solo dei primi 128 simboli che sono gli stessi**

- Si codifichi in esadecimale il seguente testo (si usi nuovamente l'Extended ASCII come tabelle di codifica)
 - “Quando sarò grande vorrò essere un direttore oppure un millepiedi” (ctr. Ralph Winchester da “I Simpson”)

```
2251 7561 6E64 6F20 7361 72F2 2067 7261 6E64 6520
766F 7272 9520 6573 7365 7265 2075 6E20 6469 7265
7474 6F72 6520 6F70 7075 7265 2075 6E20 6D69 6C6C
6570 6965 6469 2220 2863 7472 2E20 5261 6C70 6820
5769 6E63 6865 7374 6572 2064 6120 2249 2053 696D
7073 6F6E 2229
```

- Si può codificare quest'ultimo testo semplicemente in ASCII? **No** perchè il simbolo “ò” non è presente nell'originale tabella ASCII su 7 bit, ma solo sulla Extended-ASCII ad 8 bit (che corrisponde al codice 149_{10} ovvero 95_{16})