

Aritmetica dei Calcolatori 1

Laboratorio di Architettura

23 marzo 2012

1 Sistema di numerazione

- sistema posizionale

2 rappresentazione binaria

- cambio di base
- basi potenze di 2

3 Esercitazione

4 Esercitazione

Sistema di numerazione posizionale

Il sistema di numerazione che usiamo è in **base dieci** e **posizionale**:

$$2454 = 2 * 1000 + 4 * 100 + 5 * 10 + 4$$

$$2454 = 2 * 10^3 + 4 * 10^2 + 5 * 10^1 + 4 * 10^0$$

Cosa significa?

- utilizziamo **cifre** da 0 a 9
- la stessa cifra assume un **peso** diverso a seconda della posizione

In generale, fissata una base **B**

- utilizziamo **B simboli**
- un numero si esprime come **somme di potenze** della base

Cambio di base: da 10 a B

Dato un numero N in base 10, le cifre di N in base B sono i resti delle divisioni successive di N per la base B .

Un esempio: $(28)_{10} = (11100)_2 = (103)_5$

$$28/2=14 \quad \text{resto } 0$$

$$14/2= 7 \quad \text{resto } 0$$

$$7/2= 3 \quad \text{resto } 1$$

$$3/2= 1 \quad \text{resto } 1$$

$$1/2= 0 \quad \text{resto } 1$$

$$28/5=5 \quad \text{resto } 3$$

$$5/5=1 \quad \text{resto } 0$$

$$1/5=0 \quad \text{resto } 1$$

La cifra meno significativa è il primo resto calcolato!

Cambio di base: da B a 10

Dato un numero di C cifre in base B, calcoliamo la sua rappresentazione in base 10 come:

$$(N)_{10} = \sum_0^{C-1} \sigma_i B^i$$

dove σ_i è il valore in base 10 della i -esima cifra in base B.

Un esempio:

$$\begin{array}{r} (11100)_2 = 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 0 * 2^0 = (28)_{10} \\ (103)_5 = 1 * 5^2 + 0 * 5^1 + 3 * 5^0 = (28)_{10} \end{array}$$

Rappresentazione binaria senza segno

Il numero $(28)_{10}$ è quindi rappresentato, utilizzando 32 bit:

00000000000000000000000000000000111100

MSB LSB

MSB = *most significant bit*

LSB = *least significant bit*

Considerando solo i **numeri positivi**, possiamo rappresentare l'intervallo $[0; 2^{32} - 1]$, cioè $[0; 4\,294\,967\,295]$

$$(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000)_2 = (0)_{10}$$

$$(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001)_2 = (1)_{10}$$

$$(0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010)_2 = (2)_{10}$$

...

...

$$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101)_2 = (4\,294\,967\,293)_{10}$$

$$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110)_2 = (4\,294\,967\,294)_{10}$$

$$(1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111)_2 = (4\,294\,967\,295)_{10}$$

Osservazioni

Per passare da un numero in **base 2** a una base 2^p , possiamo sfruttare il legame tra le due basi: raggruppiamo le cifre binarie a gruppi di **p**, partendo dal bit meno significativo, poi sostituiamo ogni gruppo di cifre con il relativo valore decimale.

Un esempio:

$$\begin{aligned}
 B = 4 &= 2^2 \\
 (11100)_2 &= 01 \quad 11 \quad 00 \\
 &= 1 \quad 3 \quad 0 = (130)_4
 \end{aligned}$$

$$\begin{aligned}
 B = 8 &= 2^3 \\
 (111100)_2 &= 011 \quad 100 \\
 &= 3 \quad 4 = (34)_8
 \end{aligned}$$

$$\begin{aligned}
 B = 16 &= 2^4 \\
 (111100)_2 &= 0001 \quad 1100 \\
 &= 1 \quad C = (1C)_{16}
 \end{aligned}$$

Quando finiscono
le cifre?

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

⋮

Esercitazione 1/2

Implementare le funzioni:

- 1 void `dec2bin`(unsigned int `dec`, short int `bin`[32])
Dato un intero senza segno `dec`, memorizza sua rappresentazione in base 2 in `bin`. Il bit meno significativo è in `bin`[0].
- 2 unsigned int `bin2dec`(short int `bin`[32])
Dato un vettore di interi (0 o 1) `bin`, calcola la rappresentazione decimale e la ritorna.
- 3 void `dec2oct`(unsigned int `dec`, short int `oct`[11])
Dato un intero senza segno `dec`, memorizza sua rappresentazione in base 8 in `oct`. La cifra ottale meno significativa è in `oct`[0].
- 4 unsigned int `oct2dec`(short int `oct`[11])
Dato un vettore di interi (0,...,7) `oct`, calcola la rappresentazione decimale e la ritorna.

continua...

Esercitazione 2/2

Implementare le funzioni:

- 1 void **dec2hex**(unsigned int **dec**, short int **hex**[8])
Dato un intero senza segno **dec**, memorizza sua rappresentazione in base 16 in **hex**. La cifra esadecimale meno significativa è in **hex**[0].
- 2 unsigned int **hex2dec**(short int **hex**[8])
Dato un vettore di interi (0,...,F) **hex**, calcola la rappresentazione decimale e la ritorna.

Requisiti

Requisiti fondamentali:

- Non è consentito l'uso di **pow** o altre funzioni per l'elevamento a potenza,
- Il prototipo delle funzioni non va cambiato per nessun motivo!
- La stampa a video delle rappresentazioni binaria, ottale e esadecimale deve riportare la posizione delle cifre da MSB a LSB:

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
<i>MSB</i>																														<i>LSB</i>																	

0	0	0	0	0	0	0	0	0	3	4
<i>MSB</i>									<i>LSB</i>	

0	0	0	0	0	0	1	C
<i>MSB</i>				<i>LSB</i>			

Test finale

Come test per le funzioni implementate, si scriva una funzione `main()` che chiami tutte le funzioni di conversione realizzate, con almeno **cinque esempi** di input e relativo output per ciascuna funzione. Ad ogni operazione svolta si stampi a video il valore dei dati in input e della relativa conversione in output. Ad esempio:

```
me@mylaptop:~\ ./es1
(      75) Base 10 = (00000000000000000000000000001001011) Base 2
(00000000000000000000000000001001011) Base 2 = (      75) Base 10

(      27) Base 10 = (000000000000000000000000000011011) Base 2
(000000000000000000000000000011011) Base 2 = (      27) Base 10

(      18) Base 10 = (000000000000000000000000000010010) Base 2
(000000000000000000000000000010010) Base 2 = (      18) Base 10

(     4586) Base 10 = (000000000000000000001000111101010) Base 2
(000000000000000000001000111101010) Base 2 = (     4586) Base 10

(      345) Base 10 = ...
...
```

Consegna

L'esercitazione da consegnare è formata da:

- 1 un file in linguaggio C di nome `es1.c`, adeguatamente commentato
- 2 un file in formato **testo** di nome `es1.out` contenente la cattura dell'output del programma di test Usate la redirectione verso file:

```
me@mylaptop:~$ ./es1 > es1.out
```

Oppure copiate l'output e incollate su file di testo.

- NON usate la cattura di immagini da schermo: copiate ed incollate usando un editor di testo
- NON includete file BINARI o creati con WORD, EXCEL...
- NON verranno considerate mail con piu' di 2 file allegati!

Spedite tutto a arc1@fe.infn.it entro le ore 23:59:59 di mercoledì 4 aprile.

L'oggetto della mail **deve** essere nella forma:

LAB1-N#**esercitazione**-#**gruppo**

(es: LAB1-N1-99)