
INFORMATICA DI BASE

*Sistemi di numerazione:
binario, ottale ed esadecimale*

Prof. Sebastiano Battiato
Dipartimento di Matematica e Informatica
Università di Catania
e-mail : { [battiato](mailto:battiato@dmf.unict.it) } @dmf.unict.it

Il sistema di numerazione posizionale decimale

Nella numerazione posizionale ogni cifra del numero assume un valore in funzione della “*posizione*”:

221 in notazione compatta, cioè

$$2 \times 100 + 2 \times 10 + 1 \times 1$$

ovvero, con la notazione esplicita

$$2 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

Sistema posizionale

- Ogni numero si esprime come la somma dei prodotti di ciascuna cifra per la base elevata all'esponente che rappresenta la *posizione* della cifra:

$$221 = 2 \times 10^2 + 2 \times 10^1 + 1 \times 10^0$$

Sistema posizionale (cont.)

- La *notazione posizionale* può essere usata con qualunque base creando così differenti sistemi di numerazione.
- Per ogni base di numerazione si utilizza un numero di cifre uguale alla base.
- In informatica si utilizza prevalentemente la numerazione:
- binaria,
 - ottale,
 - esadecimale.
- Il sistema di numerazione romano non è posizionale:
- Ad esempio, XIII vs. CXII.

Sistema di numerazione decimale

➤ La numerazione *decimale* utilizza una notazione posizionale basata su 10 cifre (da 0 a 9) e sulle potenze di 10.

- Il numero 234 può essere rappresentato esplicitamente come:

$$2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

Sistema di numerazione binario

➤ Il sistema di numerazione *binario* utilizza una notazione posizionale basata su 2 cifre (0 e 1) e sulle potenze di 2.

- Il numero 1001 può essere rappresentato esplicitamente come:

$$\begin{aligned} 1001_2 &= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 9_{10} \end{aligned}$$

Sistema di numerazione ottale

➤ Il sistema di numerazione *ottale* utilizza una notazione posizionale basata su 8 cifre (da 0 a 7) e sulle potenze di 8.

- Il numero **534** può essere rappresentato esplicitamente come:

$$534_8 = 5 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 = 348_{10}$$

Sistema di numerazione esadecimale

➤ La numerazione *esadecimale* utilizza una notazione posizionale basata su 16 cifre (da 0 a 9 ed i caratteri A, B, C, D, E, F) e sulle potenze di 16.

- Il numero **B7FC**₁₆ può essere rappresentato esplicitamente come:

$$\begin{aligned} (11) \times 16^3 + 7 \times 16^2 + (15) \times 16^1 + (12) \times 16^0 \\ = 47100_{10} \end{aligned}$$

Conversione da base n a base 10

- Per convertire un numero da una qualunque base alla base 10 è sufficiente rappresentarlo esplicitamente:

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

$$710_8 = 7 \times 8^2 + 1 \times 8^1 + 0 \times 8^0 = 456_{10}$$

$$A51_{16} = (10) \times 16^2 + 5 \times 16^1 + 1 \times 16^0 = 2641_{10}$$

Conversione da base 10 a base n

- Per convertire un numero ad una base n qualsiasi occorre trovare tutti i resti delle successive divisioni del numero per la base n .

- Come esempio si vuole trovare il valore binario del numero **210**
 - Basterà dividere 210 per la base 2,
 - ...

Conversione da base 10 a base 2

210	2	resto	0
105	2		1
52	2		0
26	2		0
13	2		1
6	2		0
3	2		1
1	2		1



➤ Leggendo la sequenza dei resti dal basso verso l'alto, si ottiene il numero:

11010010₂

Elementi di Informatica, pag. 11

Verifica di correttezza

➤ Per una verifica di correttezza basta riconvertire il risultato alla base 10:

$$11010010_2 = 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 +$$

$$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 +$$

$$1 \times 2^1 + 0 \times 2^0 = 210_{10}$$

Elementi di Informatica, pag. 12

Costruzione dei numeri binari

➤ Per costruire la successione dei numeri binari si può seguire il seguente schema:

0	0	0	0	=	0
0	0	0	1	=	1
0	0	1	0	=	2
0	0	1	1	=	3
0	1	0	0	=	4
0	1	0	1	=	5
0	1	1	0	=	6
0	1	1	1	=	7

Elementi di Informatica, pag. 13

I primi 32 numeri binari

0	0	0	0	0	0	0	0	=	0
0	0	0	0	0	0	0	1	=	1
0	0	0	0	0	0	1	0	=	2
0	0	0	0	0	0	1	1	=	3
0	0	0	0	0	1	0	0	=	4
0	0	0	0	0	1	0	1	=	5
0	0	0	0	0	1	1	0	=	6
0	0	0	0	0	1	1	1	=	7

0	0	0	0	1	0	0	0	=	8
0	0	0	0	1	0	0	1	=	9
0	0	0	0	1	0	1	0	=	10
0	0	0	0	1	0	1	1	=	11
0	0	0	0	1	1	0	0	=	12
0	0	0	0	1	1	0	1	=	13
0	0	0	0	1	1	1	0	=	14
0	0	0	0	1	1	1	1	=	15

Elementi di Informatica, pag. 14

I primi 32 numeri binari (cont.)

0	0	0	1	0	0	0	0	= 16
0	0	0	1	0	0	0	1	= 17
0	0	0	1	0	0	1	0	= 18
0	0	0	1	0	0	1	1	= 19
0	0	0	1	0	1	0	0	= 20
0	0	0	1	0	1	0	1	= 21
0	0	0	1	0	1	1	0	= 22
0	0	0	1	0	1	1	1	= 23

0	0	0	1	1	0	0	0	= 24
0	0	0	1	1	0	0	1	= 25
0	0	0	1	1	0	1	0	= 26
0	0	0	1	1	0	1	1	= 27
0	0	0	1	1	1	0	0	= 28
0	0	0	1	1	1	0	1	= 29
0	0	0	1	1	1	1	0	= 30
0	0	0	1	1	1	1	1	= 31

Elementi di Informatica, pag. 15

Operazioni binarie

$$\begin{array}{r} 10110101+ \\ 1000110 = \\ \hline 11111011 \end{array}$$

$$\begin{array}{r} 00110011+ \\ 00111000 = \\ \hline 01101011 \end{array}$$

Elementi di Informatica, pag. 16

Operazioni binarie (cont.)

1101 x	10011 x
11 =	10 =
<hr/>	<hr/>
1101	00000
1101	10011
<hr/>	<hr/>
100111	100110

Elementi di Informatica, pag. 17

Esercizi

Eseguire le seguenti operazioni direttamente in binario, convertire in decimale e verificare il risultato:

- 110000 + 1001010;
- 1001010 + 1111111 + 10;
- 100110 x 111100;
- 001001 x 111.

Elementi di Informatica, pag. 18

INFORMATICA DI BASE

Rappresentazione delle informazioni :

La codifica dei testi

Prof. Sebastiano Battiato

Dipartimento di Matematica e Informatica

Università di Catania

e-mail : { [battiato](mailto:battiato@dmf.unict.it) } @dmf.unict.it

Rappresentazione dei caratteri

➤ Cos'è un carattere ?

- Si tratta di un simbolo, in qualche modo astratto.

- Per esempio una “A” è la rappresentazione grafica convenzionale (detta anche *glifo*) del concetto di carattere “a maiuscola”.

➤ Dobbiamo trovare una “convenzione” con cui realizzare una rappresentazione comprensibile al computer dei caratteri.

Codifica binaria

➤ Per poter rappresentare le informazioni è necessario utilizzare *sequenze* di bit.

- Utilizzando due bit si possono rappresentare quattro informazioni diverse:

00 01 10 11

➤ Il processo che fa corrispondere ad una informazione una configurazione di bit prende il nome di *codifica dell'informazione*.

Sequenze di bit

Numero di bit nella sequenza	Informazioni rappresentabili
2	4
3	8
4	16
5	32
6	64
7	128
8	256

In generale, con n bit si possono rappresentare 2^n differenti informazioni.

➤ Un gruppo di 8 bit viene denominato Byte.

- Corrisponde ad un carattere.
- Unità di misura della capacità di memoria.

➤ Si utilizzano i multipli del Byte:

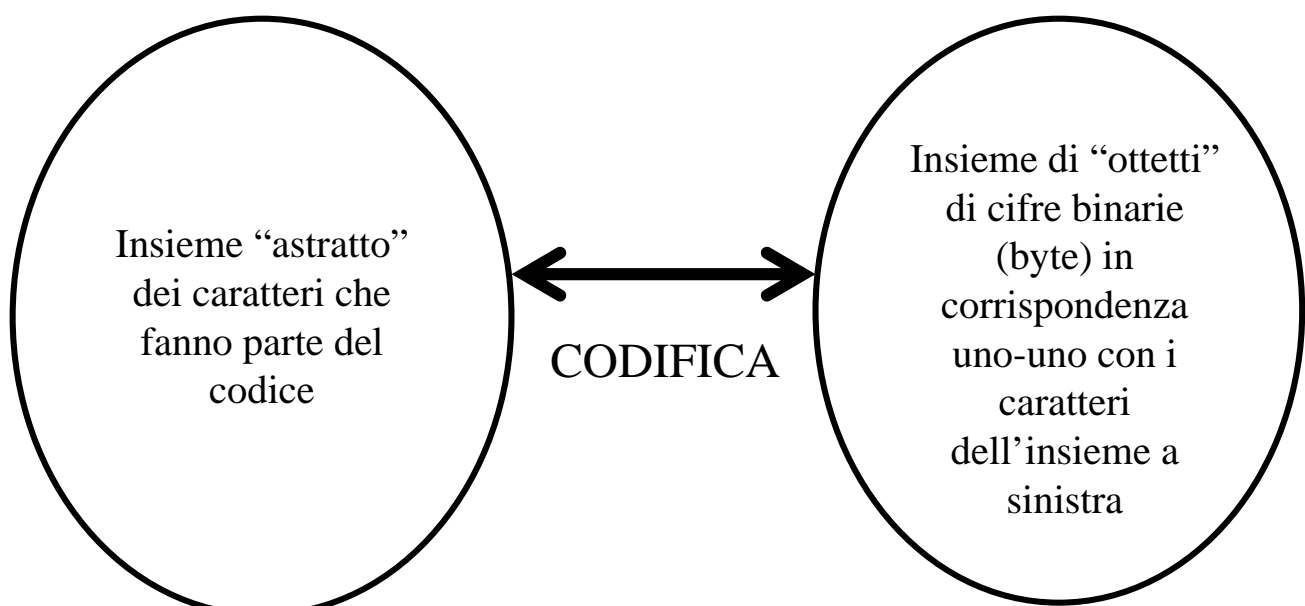
- | | | | | |
|--------|----|----------|------------------|-------------|
| • Kilo | KB | 2^{10} | ~ un migliaio | (1024) |
| • Mega | MB | 2^{20} | ~ un milione | (1024x1024) |
| • Giga | GB | 2^{30} | ~ un miliardo | (1MBx1024) |
| • Tera | TB | 2^{40} | ~ mille miliardi | (1GBx1024) |

I caratteri utilizzati nella comunicazione scritta

- 52 lettere alfabetiche maiuscole e minuscole
- 10 cifre (0, 1, 2, ..., 9)
- Segni di punteggiatura (, . ; : ! ” ? ’ ^ \ ...)
- Segni matematici (+, -, ×, ±, {, [, >, ...)
- Caratteri nazionali (à, è, ì, ò, ù, ç, ñ, ö, ...)
- Altri segni grafici (©, ←, ↑, ⚡, @, € ...)
- **In totale 220 caratteri circa.**

- Si pone quindi la necessità di codificare in numeri binari almeno 220 caratteri.
- La sequenza di bit necessaria a rappresentare 220 simboli deve essere composta da 8 bit e prende il nome di **CODICE**.

La fase di codifica



Il codice ASCII

.....			0011 0000	48	0
0100 0001	65	A	0011 0001	49	1
0100 0010	66	B	0011 0010	50	2
0100 0011	67	C	0011 0011	51	3
.....				
0101 1000	88	X	0011 1010	58	:
0101 1001	89	Y	0011 1011	59	;
0101 1010	90	Z	0011 1100	60	<
.....			0011 1101	61	=
0110 0001	97	a		
0110 0010	98	b	1010 0100	164	ñ
0110 0011	99	c	1000 0111	135	ç

American
Standard Code
for Information
Interchange

Elementi di Informatica, pag. 27

Il codice ASCII (cont.)

➤ I caratteri ASCII da 0 a 127:

- I primi 32 (numerati da 0 fino a 31) sono “caratteri di controllo” *non stampabili*,
- I successivi 95 simboli (numerati da 32 fino a 126) sono caratteri *stampabili*,
- Il 128-esimo simbolo è ancora un “carattere di controllo” *non stampabile*.

Elementi di Informatica, pag. 28

Dove trovo le tabelle ASCII ?

➤ Andate su un motore di ricerca (per esempio www.google.it) e digitate “tabella ASCII”.

- Un ottimo sito in italiano è:

- <http://www.cesit.unifi.it/online/principi/asciistd.html>

- Un buon sito in inglese è invece:

- <http://www.ibilce.unesp.br/courseware/datas/data1.htm>

Elementi di Informatica, pag. 29

Tabella ASCII in notazione binaria: la concatenazione del ‘nibble’ di riga e di quello di colonna dà il codice ASCII in binario.

	0000 0	0001 1	0010 2	0011 3	0100 4	0101 5	0110 6	0111 7	1000 8	1001 9	1010 10	1011 11	1100 12	1101 13	1110 14	1111 15
0000 0																
0001 1																
0010 2	spazio	!	“	#	\$	%	&	‘	()	*	+	,	-	.	/
0011 3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100 4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101 5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110 6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111 7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Elementi di Informatica, pag. 30

Tabella ASCII in notazione decimale: la somma dell'indice di riga e di colonna dà il codice ASCII in decimale.

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
16																
32	spazio	!	“	#	\$	%	&	'	()	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Elementi di Informatica, pag. 31

Sequenze di caratteri ASCII

Dividendo la sequenza in gruppi di byte è possibile risalire ai singoli caratteri:

01101001 01101100 00100000 01010000 01001111 00101110

01101001 01101100 00100000 01010000 01001111 00101110

i

l

P

O

.

Elementi di Informatica, pag. 32

Esempi di sequenze

➤ “Computer” in ASCII diventa:

- C=67=01000011, o=111=01101111,
 m=109=001101101, p=112=01110000,
 u=117=01110101, t=116=01110100,
 e=101=01100101, r=114=01110010.

– 01000011- 01101111- 01101101- 11100000-01110101-01110100-01100101- 01110010

➤ Esercizio :

- *Scrivere “ASCII” in decimale ed in binario.*

Numeri e codice ASCII

➤ Con il codice ASCII è possibile rappresentare i numeri come sequenza di caratteri. Ad esempio il numero 234 sarà rappresentato come:

00110010 00110011 00110100

2

3

4

➤ Con questo tipo di rappresentazione *non* è possibile effettuare operazioni aritmetiche.

Rappresentazione di dati alfabetici

Codifiche standard:

- **ASCII**, 8 bit per carattere, rappresenta 256 caratteri.
- **UNICODE**, 16 bit per carattere
 - ASCII e caratteri etnici ($2^{16} = 65.536$ simboli).

➤ Codifiche proprietarie:

- **MSWindows**, 16 bit per carattere
 - simile ad UNICODE.

Elementi di Informatica, pag. 35

Dieci dita e qualche tasto...

➤ La mia tastiera ha meno di cento tasti.

- Come ottenere tutti i simboli desiderati?
 - Usando combinazioni di tasti.
 - ❖ Per esempio, <Shift><tasto> dà la versione maiuscola.
 - Digitando la combinazione :
 - ❖ <Alt><codice ASCII in notazione decimale>
 - In questo caso bisogna usare il *tastierino numerico* per inserire il codice !

Elementi di Informatica, pag. 36

Rappresentare “testi” nel computer

➤ Cos'è un testo ?

- una sequenza ordinata di “caratteri” .
 - Esempio : `abgx76 6&&&%%””0??`
- una sequenza ordinata di caratteri *formattati*.

– Esempio : **AA***A***x**`b()`... **BB**

➤ È importante notare che, a parte la superficiale similarità, i due concetti definiti sopra sono differenti.

- Per questo richiedono rappresentazioni diverse.

Elementi di Informatica, pag. 37

Testi formattati

ciao **Ciao** ciao *ciao* **ciao** χ iao ciao
ciao ciao ciao ci ao

Sottolineato

Corsivo

Grassetto

Grassetto corsivo

Elementi di Informatica, pag. 38

INFORMATICA DI BASE

Rappresentazione delle informazioni :

La codifica dei numeri

Prof. Sebastiano Battiato

Dipartimento di Matematica e Informatica

Università di Catania

e-mail : { [battiato](mailto:battiato@dmf.unict.it) } @dmf.unict.it

Rappresentazione dei numeri

- All'interno dei computer, a causa dei vincoli tecnologici, per rappresentare qualsiasi tipo di numero, si *utilizzano sempre un numero fisso di cifre binarie.*
- Su tutti i computer si utilizzano:
 - 16 bit (2 byte)**
 - 32 bit (4 byte)**
- In alcuni casi si può arrivare a **64 bit (8 byte)** o più a seconda del tipo di processore.

Rappresentazione dei numeri (cont.)

- Tutti i numeri vengono distinti in tre categorie:
 - **Interi senza segno** (interi positivi).
 - **Interi con segno** (interi positivi e negativi).
 - **Reali** (numeri positivi e negativi con virgola).

- Ogni categoria viene rappresentata in modo differente.

Elementi di Informatica, pag. 41

Rappresentazione con numero fisso di cifre (1)

- Per comprendere il meccanismo alla base della rappresentazione con un numero fisso di cifre partiamo da un esempio:
 - **Qual è il numero più grande rappresentabile con 4 cifre?**

In base 10: **9999**

In base 2: **1111 = 15**₁₀

In base 16: **FFFF = 65535**₁₀

In base 8: **7777 = 4095**₁₀

Elementi di Informatica, pag. 42

Rappresentazione con numero fisso di cifre (2)

➤ In generale si avrà, con n cifre:

$$b^n - 1$$

In base 10: **9999 = $10^4 - 1$**

In base 2: **1111 = $2^4 - 1$**

In base 16: **FFFF = $16^4 - 1$**

In base 8: **7777 = $8^4 - 1$**

Rappresentazione con numero fisso di cifre (3)

Quindi vale la seguente *regola* nel caso di numeri interi positivi:

Nella base di numerazione b disponendo di n cifre si possono rappresentare soltanto i numeri

da 0 a $b^n - 1$

Rappresentazione dei numeri interi senza segno

➤ Per calcolare il valore massimo ammesso occorre applicare la regola $2^n - 1$ (dove n vale 16 o 32)

- Nella rappresentazione a 16 bit i possibili valori saranno compresi tra

0 e 65.535

- Nella rappresentazione a 32 bit i possibili valori saranno compresi tra

0 e 4.294.967.295

Elementi di Informatica, pag. 45

Numeri interi con segno (1)

➤ Per rappresentare i numeri con il loro segno (interi positivi e negativi) esistono due possibili modi. Il primo è il seguente:

- Dati n bit, un bit si riserva al segno e gli altri $n - 1$ sono destinati al numero
- Ad esempio, considerando 8 bit e ponendo il primo bit a sinistra 0 per il + e 1 per il - avremo:

$$0000\ 0101_2 = +5_{10}$$

$$1000\ 0101_2 = -5_{10}$$

Elementi di Informatica, pag. 46

Numeri interi con segno (2)

- Questo tipo di rappresentazione prende il nome di:

Rappresentazione in modulo e segno

- Anche se semplice, possiede però un grosso difetto: esistono due zeri.

1	0	0	0	0	0	1	1	=	-3
1	0	0	0	0	0	1	0	=	-2
1	0	0	0	0	0	0	1	=	-1
1	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
0	0	0	0	0	0	1	0	=	+2
0	0	0	0	0	0	1	1	=	+3

Elementi di Informatica, pag. 47

Numeri interi con segno (3)

- Utilizzando n bit e riservandone uno al segno, l'applicazione della formula precedente porterà:

da $-(2^{n-1} - 1)$ a $2^{n-1} - 1$

(dove n vale al solito 16 o 32).

Elementi di Informatica, pag. 48

Numeri interi con segno (4)

➤ Seguendo il ragionamento precedente, i possibili valori nel caso di 16 bit saranno quindi compresi tra

$$-32.767 \text{ e } +32.767$$

➤ Nel caso dei 32 bit si avrà:

$$-2.147.483.647 \text{ e } +2.147.483.647$$

Numeri interi con segno (5)

➤ Il secondo modo per rappresentare i numeri con il loro segno (interi positivi e negativi) è quello del *complemento a due*:

- Con n bit si possono rappresentare i numeri da

$$-2^{n-1} \text{ a } 2^{n-1} - 1$$

- Dato un numero composto da n bit, la rappresentazione in complemento a due si ottiene invertendo gli 1 in 0 e gli 0 in 1, e poi sommando 1 al risultato ottenuto.

Numeri interi con segno (6)

$$n = 3$$

1	0	0	=	-4
1	0	1	=	-3
1	1	0	=	-2
1	1	1	=	-1
0	0	0	=	0
0	0	1	=	+1
0	1	0	=	+2
0	1	1	=	+3

- È presente un solo zero!
- Tutti i numeri positivi cominciano con 0 mentre quelli negativi cominciano con 1.
- Complementando si passa dal valore positivo allo stesso valore negativo e viceversa (ad esclusione del -4).
 - Codificare -3 in complemento a due:
 - si considera +3 in binario,
 - si invertono le cifre : 100,
 - si somma 1, ottenendo 101;
 - e viceversa da 101 si ottiene 011.

Elementi di Informatica, pag. 51

L'overflow

- Per questioni tecnologiche tutti i computer, senza alcuna eccezione, trattano i numeri sempre con un numero fisso di cifre binarie (ad esempio 16, 32 o più).
- Quando l'elaboratore esegue un'operazione il cui risultato eccede il numero di cifre permesso, la computazione si arresta immediatamente e viene segnalato l'errore di **OVERFLOW**.

Elementi di Informatica, pag. 52

L'overflow (cont.)

- Ad esempio se la rappresentazione è a 32 bit senza segno e si vuole eseguire la seguente operazione:

$$\begin{array}{r} 3.000.000.000 + \\ \underline{2.000.000.000} = \end{array}$$

***** Errore di **OVERFLOW**

La computazione si arresta immediatamente.

Elementi di Informatica, pag. 53

Rappresentazione dei numeri con virgola

- I numeri con la virgola vengono rappresentati mediante la *notazione scientifica* in virgola mobile.
- Anche i numeri interi possono sempre essere espressi come numeri con virgola attraverso la notazione scientifica.

Elementi di Informatica, pag. 54

Rappresentazione dei numeri con virgola (cont.)

Numero	Virgola mobile	Parte decimale (mantissa)	Exp
250	$0,25 \times 10^3$	25	3
-83,76	$-0,8376 \times 10^2$	-8376	2
0,05	$0,5 \times 10^{-1}$	5	-1
55.640.350	$0,5564035 \times 10^8$	5564035	8

Elementi di Informatica, pag. 55

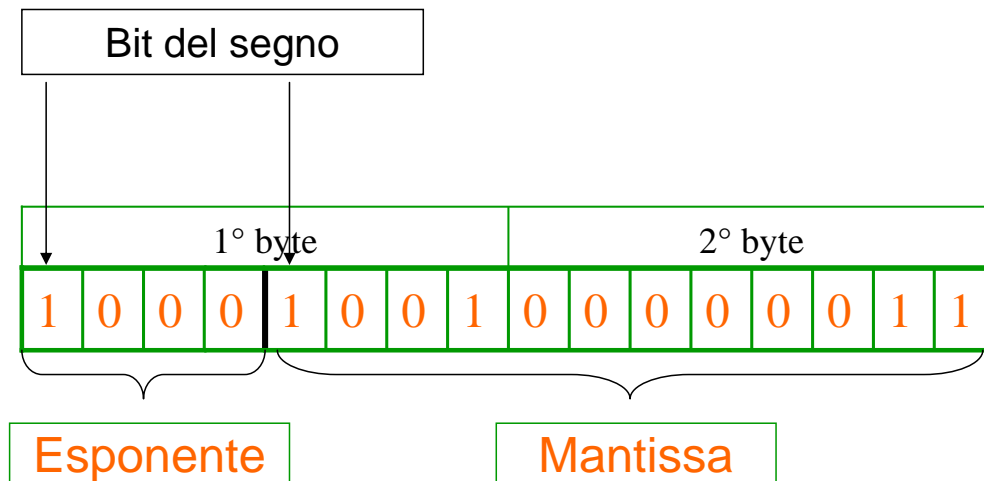
Rappresentazione dei numeri con virgola (cont.)

Parte decimale (mantissa)	Exp
25	3
-8376	2
5	-1
5564035	8

Con la notazione scientifica in virgola mobile ogni numero viene memorizzato solo come mantissa (parte decimale senza lo 0 e la virgola) e con l'esponente (senza la base): si risparmia spazio.

Elementi di Informatica, pag. 56

Rappresentazione dei numeri con virgola (cont.)



Elementi di Informatica, pag. 57

INFORMATICA DI BASE

Rappresentazione delle informazioni :

Codifica di dati multimediali

Prof. Sebastiano Battiato

Dipartimento di Matematica e Informatica

Università di Catania

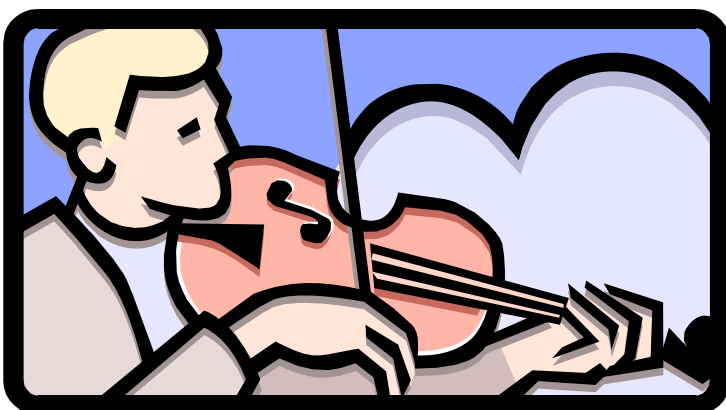
e-mail : { [battiato](mailto:battiato@dm.unict.it) } @dm.unict.it

Ma il mondo non è tutto “scritto” !

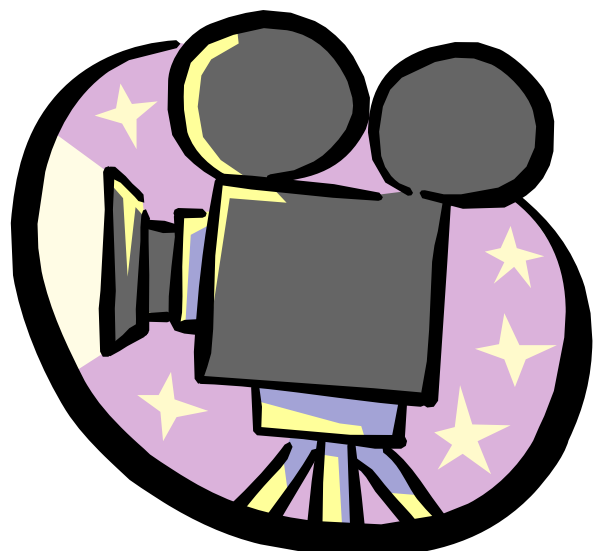
- I caratteri alfanumerici non costituiscono le uniche informazioni utilizzate dagli elaboratori.
 - Le applicazioni *multimediali* utilizzano ed elaborano informazioni contenenti:
 - immagini,
 - suoni,
 - filmati.

Elementi di Informatica, pag. 59

La codifica dei dati multimediali



L'informazione per gli umani ha un carattere analogico....



Elementi di Informatica, pag. 60

In termini tecnici si tratta di “SEGNALI”

➤ Un segnale analogico si può:

- **CAMPIONARE**

- Per un suono: misurare l'intensità ogni centesimo di secondo.

- Per una immagine : misurare i colori ogni millimetro quadrato.

- **RAPPRESENTARE con un numero.**

La codifica delle immagini (1)

➤ Esistono numerose tecniche che vengono utilizzate per la memorizzazione e l'elaborazione di un'immagine.

➤ Consideriamo un'immagine in bianco e nero, senza ombreggiature o livelli di chiaroscuro.

La codifica delle immagini (2)

L'immagine viene suddivisa mediante una griglia formata da righe orizzontali e verticali a distanza costante.



Elementi di Informatica, pag. 63

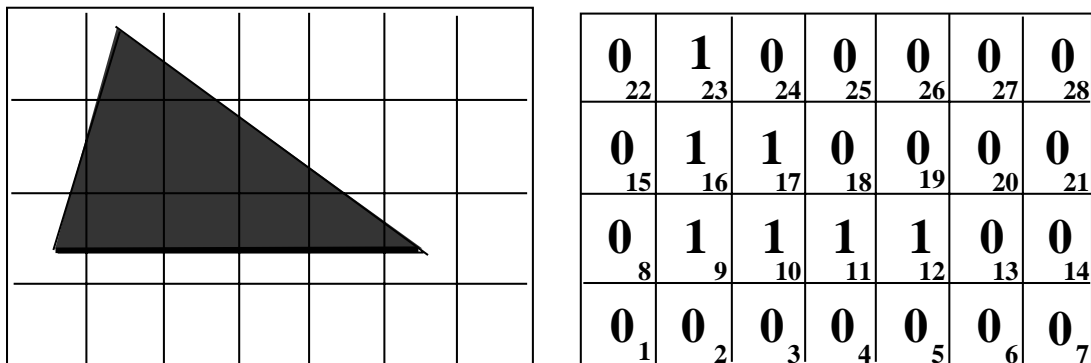
La codifica delle immagini (3)

- Ogni quadrato prende il nome di **pixel** (**picture element**) e viene codificato in binario secondo la seguente convenzione:
- Il simbolo 0 viene utilizzato per la codifica di un pixel corrispondente ad un quadrato in cui il bianco è predominante.
 - Il simbolo 1 viene utilizzato per la codifica di un pixel corrispondente ad un quadratino in cui il nero è predominante.

Elementi di Informatica, pag. 64

La codifica delle immagini (4)

- Per convenzione la griglia dei pixel è ordinata dal basso verso l'alto e da sinistra verso destra.



La figura sarà rappresentata dalla stringa binaria:

000000 0111100 0110000 0100000

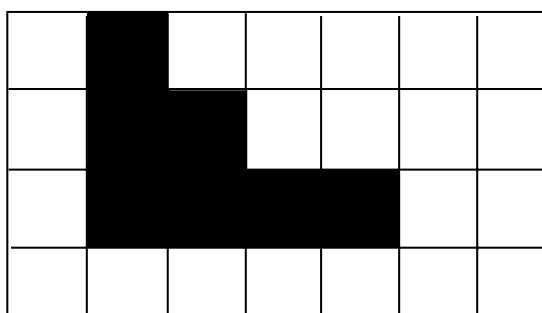
La codifica delle immagini (5)

- Dato che il contorno della figura non sempre coincide con la griglia si ottiene un'approssimazione della figura originaria.

- Riconvertendo la stringa:

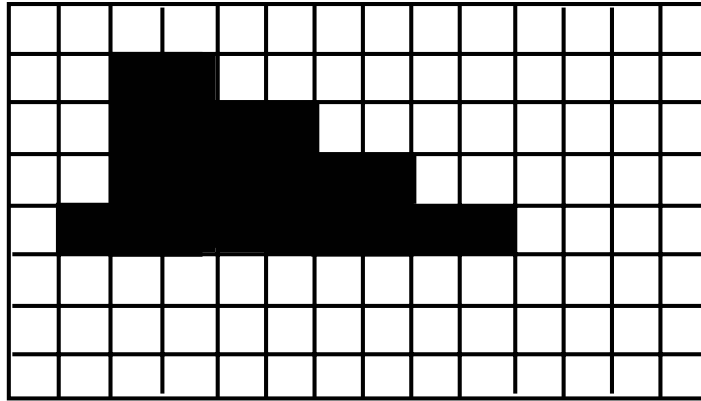
– 0000000 0111100 0110000 0100000

si avrà:



La codifica delle immagini (6)

- La rappresentazione sarà più fedele all'aumentare del numero di pixel.



- La *dimensione* dell'immagine è espressa come numero di pixel nel formato *righe x colonne*.

Elementi di Informatica, pag. 67

Gradazioni di grigio

- Per codificare immagini con diversi livelli di grigio si usa una rappresentazione binaria: ad ogni livello di grigio corrisponde una sequenza di bit.
 - Ad esempio, utilizzando quattro bit si possono rappresentare $2^4 = 16$ livelli di grigio, mentre con otto bit ne possiamo distinguere $2^8 = 256$.

Elementi di Informatica, pag. 68

Esempi di toni di grigio



1 bit
2 toni



2 bit
4 toni



8 bit
256 toni

Elementi di Informatica, pag. 69

L'uso del colore

- Il colore viene generato dalla composizione di tre *colori primari*: Red, Green, Blue (video RGB)
- Ad ogni colore primario si associa una possibile sfumatura o *gradazione* mediante un'opportuna sequenza di bit.
 - Utilizzando 2 bit per ogni colore primario si possono ottenere 4 gradazioni per ognuno di essi, cioè 64 colori.
 - Un pixel richiede quindi un byte circa (6 bit) in questa maniera.

Elementi di Informatica, pag. 70

L'uso del colore (cont.)

- Utilizzando 4 bit per ogni colore primario si ottengono 16 gradazioni per ognuno di essi, cioè 4096 colori differenti.
 - Un pixel richiede quindi circa due byte (12 bit) di informazione.
- Utilizzando 8 bit per ogni colore primario si ottengono 256 gradazioni per ognuno di essi, cioè 16,8 milioni di colori circa.
 - Un pixel richiede quindi tre byte di informazione.

Elementi di Informatica, pag. 71

Risoluzione

- Il numero di pixel per unità di area indica la “risoluzione” con cui si è campionata l’immagine.
- Il numero di pixel presenti sul video (*colonne x righe*) prende il nome di **risoluzione dello schermo**.
 - Risoluzioni tipiche sono: 640x480 800x600 1024x768
 - La dimensione dell’immagine sarà:

	16 bit	32 bit
640x480	600 KB	~ 1,2 MB
800x600	~ 938 KB	~ 1,9 MB
1024x768	~ 1,6 MB	3MB

Elementi di Informatica, pag. 72

Formati delle immagini

➤ Codifica *raster* o *bitmap*.

- Ogni punto rappresenta un'informazione indipendente.
 - GIF,
 - JPEG,
 - BMP.

➤ Codifiche *ibride* (*raster/vettoriale*).

- Ogni elemento geometrico primitivo viene specificato individualmente.
 - Postscript,
 - PDF (Portable Document Format).

Elementi di Informatica, pag. 73

Codifica di filmati video

➤ Un filmato è una sequenza di immagini statiche (dette *fotogrammi* o *frames*).

- Codifica normale (dei singoli frames).
 - 3 minuti di video con 24 frames/sec,
 - ❖ minimo 16 frame/s per non percepire i singoli fotogrammi.
 - Risoluzione singolo frame : 200x100, 16 bit/pixel.
 - Memoria necessaria: $(3*60*24) (200*100*2) \sim 165$ MB.
- Codifica differenziale.
 - È inefficiente codificare completamente ogni frame.
 - ❖ Alcuni frames si codificano interamente, altri solo nelle parti che differiscono da quelli adiacenti.

Elementi di Informatica, pag. 74

Formati video

- **MPEG** (Moving Picture Experts Group)
 - Costituisce uno standard.
 - Molto efficiente ma complesso.

- **QuickTime**
 - Proposto da Apple.
 - Meno efficiente ma più semplice.

- **Indeo - AVI**
 - Proposto da Intel, usato da MicroSoft.
 - È il più inefficiente.

Elementi di Informatica, pag. 75

Codifica di suoni

- Il segnale acustico viene digitalizzato.
 - **Dimensioni medie:**
 - un minuto di audio con qualità CD musicale stereo occupa da 1MB a 10MB a seconda della codifica impiegata.

- **Codifiche standard:**
 - WAV (MS-Windows),
 - MIDI
 - MP3

Elementi di Informatica, pag. 76

Formati sonori

➤ MIDI:

- Codifica le note e gli strumenti che devono eseguirle.
- Solo musica, non voce.
- Richiede un sintetizzatore o “campioni” per la riproduzione (non utilizzabile “direttamente”).
- Molto efficiente.

➤ MP3:

- MPEG - layer 3: variante MPEG per i suoni.
- Grande diffusione.
- Molto efficiente.

Elementi di Informatica, pag. 77

Ma quanto spazio ci vuole?

➤ I bit costano!

➤ Trasmettere i bit costa!

➤ Gli uomini però hanno linguaggi “ridondanti”

- Esempio : Dmn è mrcl d e nn c sn lzn d prgrmmzn
– È un messaggio poco chiaro ma decodificabile...
E occupa meno caratteri (e meno bit) di “Domani è mercoledì e non ci sono lezioni di programmazione”.

➤ IDEA: compressione!!!!

Elementi di Informatica, pag. 78

Fine