

## **Laboratorio di Informatica per STAN**

**a.a. 2003/2004**

Prof. Eugenio Moggi

moggi@disi.unige.it

<http://www.disi.unige.it/person/MoggiE/LI03/>

---

---

---

---

---

---

---

---

### **Prima di iniziare**

1. *Avete un computer a casa? E' connesso in rete (via modem)?*
2. *Avete attivato l'account webmail dell'universita'? Dovete attivarlo al piu' presto!*
3. *Avete gia' usato l'aula informatica di Facolta'? Ci andremo domani dopo la lezione!*
4. *Avete familiarita' con la posta elettronica?*
5. *Avete navigato in internet (Netscape, Explorer)?*
6. *Avete usato Windows e/o altri prodotti Microsoft (word, office, excel, access)?*

---

---

---

---

---

---

---

---

### **Programma del primo modulo**

1. *La struttura del calcolatore e la rappresentazione dell'informazione*
2. *Il sistema operativo (cenni)*
3. *Reti di elaboratori, Internet e le principali applicazioni (posta elettronica, www, ecc.)*
4. *Motori di ricerca e reperimento delle informazioni in rete*

---

---

---

---

---

---

---

---

**Definizione**

**Informatica**

**Scienza della rappresentazione e dell'elaborazione dell'informazione**

---

---

---

---

---

---

---

---

**L'informatica tratta ...**

**l'informazione**

la sua codifica  
le tecniche per raccoglierla, memorizzarla, distribuirla, trasformarla, ...



**il calcolatore**

il suo funzionamento, le possibilità che offre per la trasformazione dell'informazione, le tecniche di utilizzo, ...



**la comunicazione**

tra elaboratori, tra persone (mediata dal calcolatore), ...

---

---

---

---

---

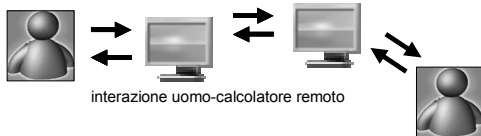
---

---

---

**Possibili interazioni ...**

interazione uomo-macchina



interazione uomo-calcolatore remoto

interazione tra utenti mediata dal calcolatore

---

---

---

---

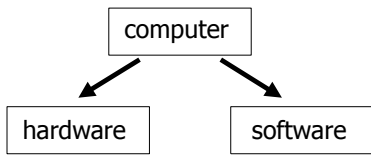
---

---

---

---

**In pillole ...**



---

---

---

---

---

---

---

**In pillole ...**

hardware

Struttura fisica del calcolatore formata da parti meccaniche, elettriche, elettroniche



---

---

---

---

---

---

---

**In pillole ...**

software

Componente del calcolatore costituita dai programmi di base e dai programmi applicativi per la gestione e l'uso del sistema



---

---

---

---

---

---

---

## La rappresentazione dell'informazione

---

---

---

---

---

---

---

### Tipi di informazione

- Esistono vari tipi di informazione, di natura e forma diversa, così come rappresentazioni diverse della stessa informazione
- Il calcolatore memorizza ed elabora informazioni che devono pertanto essere rappresentate in una forma gestibile
- Rappresentazione **digitale**

---

---

---

---

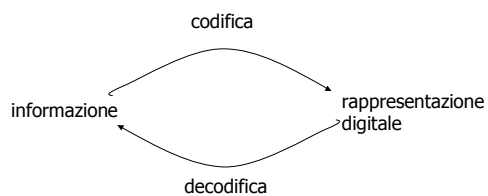
---

---

---

### Tipi di informazione

Semplificando un po'....



---

---

---

---

---

---

---

**Codifica dell'informazione**

- **bit** (binary digit - cifra binaria): **0** o **1**
  - Per poter rappresentare un numero maggiore di informazioni si usano **sequenze** di bit
- 00 01 10 11**
- Il processo secondo cui si fa corrispondere ad un'informazione una configurazione di bit prende il nome di **codifica dell'informazione**

---

---

---

---

---

---

---

---

**Codifica dell'informazione**

- **Esempio:** un esame può avere quattro possibili esiti
  - ottimo → 00
  - discreto → 01
  - sufficiente → 10
  - insufficiente → 11
- Con 2 bit si codificano 4 informazioni ( $2^2$ )
- Con 3 bit si codificano 8 informazioni ( $2^3$ )
- .....
- Con N bit si codificano  $2^N$  informazioni

---

---

---

---

---

---

---

---

**Codifica dell'informazione**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	1

.....



1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---

---

---

---

---

---

---


---

---

**Codifica dell'informazione**

*Di solito si usano i multipli del byte*

Kilo	KB	$2^{10}$	(~ un migliaio, 1024)
Mega	MB	$2^{20}$	(~ un milione, 1KBx1024)
Giga	GB	$2^{30}$	(~ un miliardo, 1MBx1024)
Tera	TB	$2^{40}$	(~ mille miliardi, 1TBx1024)



---

---

---

---

---

---

---

---

**Codifica dell'informazione**

- Problema inverso: quanti bit ci vogliono per rappresentare **M** informazioni diverse?

$$2^N \geq M$$

- **Esempio:** dovendo rappresentare 1.000 informazioni diverse dobbiamo avere a disposizione **N=10** bit per la codifica

$$2^{10} = 1024$$

NB. "avanzano" delle configurazioni ma non è possibile usare solo 9 bit per ogni informazione

---

---

---

---

---

---

---

---

**Codifica dei caratteri**

- Alfabeto anglosassone → per codificare ogni carattere sono sufficienti **7 bit** (ASCII standard)
- **8 bit** (ASCII esteso)
- **16 bit** (UNICODE)
- MS Windows usa un codice proprietario a 16 bit per carattere, simile ad UNICODE

ASCII = American Standard Code for Information Interchange

---

---

---

---

---

---

---

---

### Codifica dei caratteri (ASCII)

ASCII	Simb.	ASCII	Simb.	ASCII	Simb.
00000000	NUL	00001110	SO	00011100	PS
00000001	SOH	00001111	SI	00011101	OS
00000010	STX	00010000	DLE	00011110	RS
00000011	ETX	00010001	DC1	00011111	US
00000100	EOT	00010010	DC2	00100000	SF
00000101	ENO	00010011	DC3	00100001	!
00000110	ACK	00010011	DC4	00100010	"
00000111	BEL	00010101	NAK	00100011	#
00001000	BS	00010110	SYN	00100100	\$
00001001	HT	00010111	ETB	00100101	%
00001010	NL	00011000	CAN	00100110	&
00001011	VT	00011001	EM	00100111	'
00001100	NP	00011010	SUB	00101000	(
00001101	CR	00011011	ESC	00101001	)

---

---

---

---

---

---

---

---

---

---

### Codifica dei caratteri (ASCII)

ASCII	Simb.	ASCII	Simb.	ASCII	Simb.
00101010	*	00111001	9	01000111	G
00101011	+	00111010	:	01001000	H
00101100	,	00111011	;	01001001	I
00101101	-	00111100	<	01001010	J
00101110	.	00111101	=	01001011	K
00101111	/	00111110	>	01001100	L
00110000	0	00111111	?	01001101	M
00110001	1	01000000	@	01001110	N
00110010	2	01000001	A	01001111	O
00110011	3	01000010	B	01010000	P
00110100	4	01000011	C	01010001	Q
00110101	5	01000100	D	01010010	R
00110110	6	01000101	E	01010011	S
00110111	7	01000110	F	01010100	T

---

---

---

---

---

---

---

---

---

---

### Codifica delle parole

- ... e le parole? Sono sequenze di caratteri

- *Esempio: scienze ambientali*

01110011 01100011 01101001 01100101 01101110 01111010 01100101 00000000 →  
s c i e n z e

01100001 01101101 01100010 01101001 01100101 01101110 01110100 01100001 01101100 01101001  
a m b i e n t a l i

---

---

---

---

---

---

---

---

---

---

**Codifica dell'informazione: verifica**



1. Nell'alfabeto di Marte sono previsti 300 simboli; quanti bit si devono utilizzare per rappresentarli tutti?
1. Quanti byte occupa la frase "biologia marina" se la si codifica utilizzando il codice ASCII esteso?
1. Quanti byte occupa la stessa frase scritta in codice UNICODE?
1. Dati 12 bit per la codifica, quante informazioni distinte si possono rappresentare?

---

---

---

---

---

---

---

---

**Codifica delle immagini**



Pixel = picture element

---

---

---

---

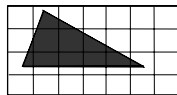
---

---

---

---

**Codifica delle immagini**



0	1	0	0	0	0	0
22	23	24	25	26	27	28
0	1	1	0	0	0	0
15	16	17	18	19	20	21
0	1	1	1	1	0	0
8	9	10	11	12	13	14
0	0	0	0	0	0	0
1	2	3	4	5	6	7

---

---

---

---

---

---

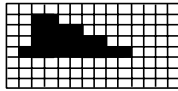
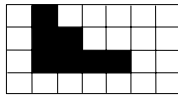
---

---



**Codifica delle immagini**

0000000011110001100000100000



---

---

---

---

---

---

---

---

**Codifica delle immagini**

- Assegnando un bit ad ogni pixel è possibile codificare solo immagini in bianco e nero
- Per codificare le immagini con diversi livelli di grigio oppure a colori si usa la stessa tecnica: per ogni pixel viene assegnata una rappresentazione binaria
- Per memorizzare un pixel non è più sufficiente un solo bit. Ad esempio, se utilizziamo 4 bit possiamo rappresentare  $2^4=16$  livelli di grigio o 16 colori diversi, mentre con 8 bit ne possiamo distinguere  $2^8=256$ , ecc.

---

---

---

---

---

---

---

---

**Codifica delle immagini: verifica**



1. Quanti byte occupa un'immagine di 100x100 pixel in bianco e nero?
1. Quanti KB occupa un'immagine di 100x100 pixel a 256 colori?
1. Se un'immagine a 16,8 milioni di colori occupa 2400 byte, da quanti pixel sarà composta?

---

---

---

---

---

---

---

---

**L'uso del colore**

- Il colore viene generato componendo 3 colori: Red, Green, Blue (RGB)
- Ad ogni colore si associa una possibile sfumatura
- Usando 2 bit per ogni colore si possono ottenere 4 sfumature per il rosso, 4 per il blu e 4 per il verde che, combinate insieme, danno origine a 64 colori diversi
- Ogni pixel per essere memorizzato richiede 6 bit

---

---

---

---

---

---

---

---

**L'uso del colore**

- Usando 8 bit per ogni colore si possono ottenere 256 sfumature per il rosso, 256 per il blu e 256 per il verde che, combinate insieme, danno origine a 16,8 milioni di colori diversi
- Ogni pixel per essere memorizzato richiede 3 byte



---

---

---

---

---

---

---

---

**Colori e risoluzione**

- Il numero di pixel presenti sullo schermo (colonne x righe) prende il nome di **risoluzione**
- Risoluzioni tipiche sono
  - 640 x 480
  - 800 x 600
  - 1024 x 768

---

---

---

---

---

---

---

---

### **Grafica bitmap**

- Le immagini codificate pixel per pixel sono dette immagini in grafica bitmap
- La grafica bitmap va bene per immagini complesse o irregolari. I formati più conosciuti sono: BITMAP (.bmp), GIF (.gif), JPEG (.jpg)

*Nelle pagine web si usano principalmente le immagini in formato GIF o JPEG (recentemente anche PNG)*



---

---

---

---

---

---

---

---

### **Grafica bitmap**

- Le immagini bitmap occupano parecchio spazio
- Esistono delle tecniche di compressione che permettono di ridurre le dimensioni
- Ad esempio, se più punti vicini di un'immagine assumono lo stesso colore, si può memorizzare la codifica del colore una sola volta e poi ricordare per quante volte deve essere ripetuta
- GIF e JPEG sono formati compressi

---

---

---

---

---

---

---

---

### **Grafica vettoriale**

- Se le immagini sono regolari si può usare una codifica di tipo **vettoriale** in cui non si specificano le informazioni di colore dei singoli pixel ma ogni elemento geometrico primitivo viene specificato individualmente
- Le immagini vengono costruite a partire dalla descrizione degli elementi che le compongono mediante un linguaggio testuale
- Spesso occupano meno spazio rispetto alle immagini bitmap

---

---

---

---

---

---

---

---

**Codifica dei filmati video**

- Un filmato è una sequenza di immagini statiche (dette fotogrammi o frame)
- Per codificare un filmato si digitalizzano i suoi fotogrammi
- Compressione: MPEG, differenza tra fotogrammi

---

---

---

---

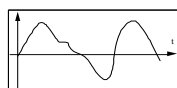
---

---

---

**Codifica dei suoni**

- Fisicamente un suono è rappresentato come un'onda che descrive la variazione della pressione dell'aria nel tempo (onda sonora)



- Sull'asse delle ascisse viene posto il tempo  $t$  e sull'asse delle ordinate la variazione della pressione corrispondente

---

---

---

---

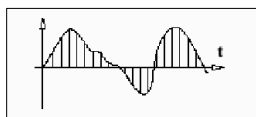
---

---

---

**Codifica dei suoni**

- Si effettuano dei campionamenti sull'onda (cioè si misura il valore dell'onda ad intervalli di tempo costanti) e si codificano in forma digitale le informazioni estratte da tali campionamenti



- La sequenza dei valori numerici ottenuta dai campioni può essere facilmente codificata

---

---

---

---

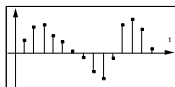
---

---

---

**Codifica dei suoni**

- Quanto più frequentemente il valore dell'onda viene campionato, tanto più precisa sarà la sua rappresentazione



- Il numero di campioni raccolti per ogni secondo definisce la frequenza di campionamento che si misura in Hertz (Hz)

---

---

---

---

---

---

---

---

**Codifica dei suoni: verifica**



1. Quanto spazio occupa un suono della durata di 10 secondi campionato a 100 Hz, in cui ogni campione occupa 4 byte?
1. Un secondo di suono campionato a 64 Hz occupa 1Kb (= 1024 bit). Quanti valori distinti si possono avere per i campioni?

---

---

---

---

---

---

---

---

**Codifica dei numeri**

- Il codice ASCII consente di codificare le cifre decimali da "0" a "9" fornendo in questo modo un metodo per la rappresentazione dei numeri
- Il numero 324 potrebbe essere rappresentato dalla sequenza di byte: 00110011 00110010 00110100  
                  3                  2                  4
- Questa rappresentazione non è efficiente e, soprattutto, non è adatta per eseguire le operazioni aritmetiche sui numeri

---

---

---

---

---

---

---

---

**Codifica dei numeri: il sistema decimale**

- Sistema posizionale in cui ogni cifra di un numero assume un valore che dipende dalla sua posizione

$$365 = 3 \times 100 + 6 \times 10 + 5 \times 1$$
$$365 = 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$$

*Si deve fare la somma dei prodotti di ciascuna cifra moltiplicata per la base elevata all'esponente che rappresenta la posizione della cifra stessa (partendo da 0)*



---

---

---

---

---

---

---

**Codifica dei numeri**

- La notazione posizionale può essere usata con qualunque base creando così sistemi di numerazione diversi
- Per ogni sistema di numerazione si usa un numero di cifre uguale alla base

*In informatica si usano prevalentemente le numerazioni binaria (base 2), ottale (base 8) ed esadecimale (base 16)*



---

---

---

---

---

---

---

**Sistema binario**

- Utilizza una notazione posizionale basata su 2 cifre (0 e 1) e sulle potenze di 2
- Esempio:  $10011 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19$
- Esempio:  $20011 \text{ ???}$

---

---

---

---

---

---

---

### Sistema ottale

- Utilizza una notazione posizionale basata su 8 cifre (0,1, ..., 7) e sulle potenze di 8
- Esempio:  $10011_8 = 1x8^4 + 0x8^3 + 0x8^2 + 1x8^1 + 1x8^0 = 4105$
- Per evitare ambiguità si può scrivere esplicitamente la base di un numero
- Esempio:  $10011_2 \neq 10011_8 \neq 10011_{10}$

---

---

---

---

---

---

---

---

### Sistema esadecimale

- Utilizza una notazione posizionale basata su 16 cifre (0,1,2,...,9,A,B,C,D,E,F) e sulle potenze di 16
- Esempio:  
 $10011_{16} = 1x16^4 + 0x16^3 + 0x16^2 + 1x16^1 + 1x16^0 = 65553$
- Esempio:  $AAC3_{16} = 10x16^3 + 10x16^2 + 12x16^1 + 3x16^0 = 43715$



---

---

---

---

---

---

---

---

### Conversione da base 10 a base 2

- Per convertire un numero in base 2 si devono trovare i resti delle divisioni successive del numero per la base 2
- Esempio:  $210_{10}$

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



---

---

---

---

---

---

---

---

### Conversione da base 10 a base 2

- Leggendo la sequenza dal basso verso l'alto si ottiene il numero

11010010<sub>2</sub>

- Per una corretta verifica basta riconvertire il risultato alla base 10

Per le altre basi il procedimento è lo stesso, cambiando il divisore



---

---

---

---

---

---

---

### Rappresentazione dei numeri

- I numeri vengono distinti in tre categorie
  - Interi positivi
  - Interi con segno (positivi e negativi)
  - Reali (positivi e negativi con virgola)
- Ogni categoria viene rappresentata in modo diverso

---

---

---

---

---

---

---

### Numeri interi positivi

- Dobbiamo usare un numero fissato di cifre
- Esempio: qual è il numero più grande rappresentabile con 4 cifre?

in base 10 9999

in base 2 1111 (= 15<sub>10</sub>)

in base 16 FFFF (= 65535<sub>10</sub>)

in base 8 7777 (= 4095<sub>10</sub>)

---

---

---

---

---

---

---



### Numeri interi positivi

- Deciso il numero di cifre a disposizione si fissa anche il **numero massimo** rappresentabile, numeri più grandi causano problemi di **overflow**
- *Esempio: 4 cifre*

*in base 10*     $9999 + 1 = 10000_{10}$

*in base 2*     $1111 + 1 = 10000_2 (= 16_{10})$

*In base 16*  $FFFF + 1 = 10000_{16} (= 65536_{10})$

*in base 8*     $7777 + 1 = 10000_8 (= 4096_{10})$

---

---

---

---

---

---

---

---

### Numeri interi positivi

- In generale, con **n** cifre a disposizione e base **b** il più grande numero (intero positivo) rappresentabile si può esprimere come

$$b^n - 1$$

- *Esempio:*

*in base 10*     $9999 = 10^4 - 1$

*in base 2*     $1111 = 2^4 - 1$

*in base 16*     $FFFF = 16^4 - 1$

*in base 8*     $7777 = 8^4 - 1$

---

---

---

---

---

---

---

---

### Numeri interi con segno e numeri reali

- Anche in questi casi sono state definite delle tecniche per la loro codifica
- Usando queste rappresentazioni si possono fare le usuali operazioni matematiche
- Per questa parte di veda il Capitolo 2 del libro Introduzione all'Informatica, Console – Ribaud, UTET (1997)

---

---

---

---

---

---

---

---

***La rappresentazione dell'informazione***

....

indipendentemente dall'informazione di partenza si ottiene sempre una sequenza di bit



---

---

---

---

---

---

---