

Rappresentazione e codifica delle informazioni

Elaborazione dell'Informazione

Elaborare le informazioni può essere utile
per **RISOLVERE PROBLEMI**

ESEMPIO

Problema: far passare il pianoforte dalla
porta del salotto



Elaborazione dell'Informazione

PROBLEMA

far passare il pianoforte
dalla porta del salotto

PRIMA ALTERNATIVA

Accostare il pianoforte alla porta
e procedere per tentativi



Elaborazione dell'Informazione

PROBLEMA

far passare il pianoforte dalla porta del salotto

SECONDA ALTERNATIVA:

Misurare porta e pianoforte (**informazioni**)
e capire se la cosa e' possibile (**elaborazione**)



Elaborazione dell'Informazione

Qual e' il vantaggio dell'elaborazione dell'informazione per la soluzione dei problemi?

Elaborazione dell'Informazione

Qual e' il vantaggio dell'elaborazione dell'informazione per la soluzione dei problemi?

GENERALITA'

Posso usare la soluzione generale al problema per risolvere diverse istanze del problema (dove la sola differenza e' nei valori assunti dai dati)

Elaborazione automatica dell'Informazione

I computer sono in grado di compiere attività di elaborazione delle informazioni in maniera
AUTOMATICA

Perché ci interessa???

Elaborazione automatica dell'Informazione

I computer sono in grado di compiere attività di elaborazione delle informazioni in maniera
AUTOMATICA

Perché ci interessa???

- Per ridurre i **tempi** di elaborazione
 - Per aumentare l'**affidabilità**
- Per non occuparci di **compiti “noiosi”** e ripetitivi
- Per rendere **accessibili** a un maggior numero di persone competenze poco comuni

Esecutore: il computer

HARD (duro) **WARE** (merce) = ferraglia
+
SOFT (morbido) **WARE**

termini universali
(... in Francia.... materiel e logiciels)

Analogia: libro/romanzo

**Come rappresentiamo e manipoliamo
l'informazione con/per il nostro esecutore????**

Rappresentazione dell'informazione

Comunicazione necessita accordo su notazione

Nel nostro mondo....

- 21 (o 26) lettere dell'alfabeto (parole)
- 10 cifre (numeri interi)
- notazione segno/grandezza (+3, -245....)
- notazione decimale (13,45 ...)
- 7 note musicali

.....

... e nel mondo del computer????

Rappresentazione dell'informazione

Sistema di numerazione BINARIO

0 , 1

Binary digit = BIT

Bastano due cifre a rappresentare tutto quello
che ci interessa????

Rappresentazione dell'informazione

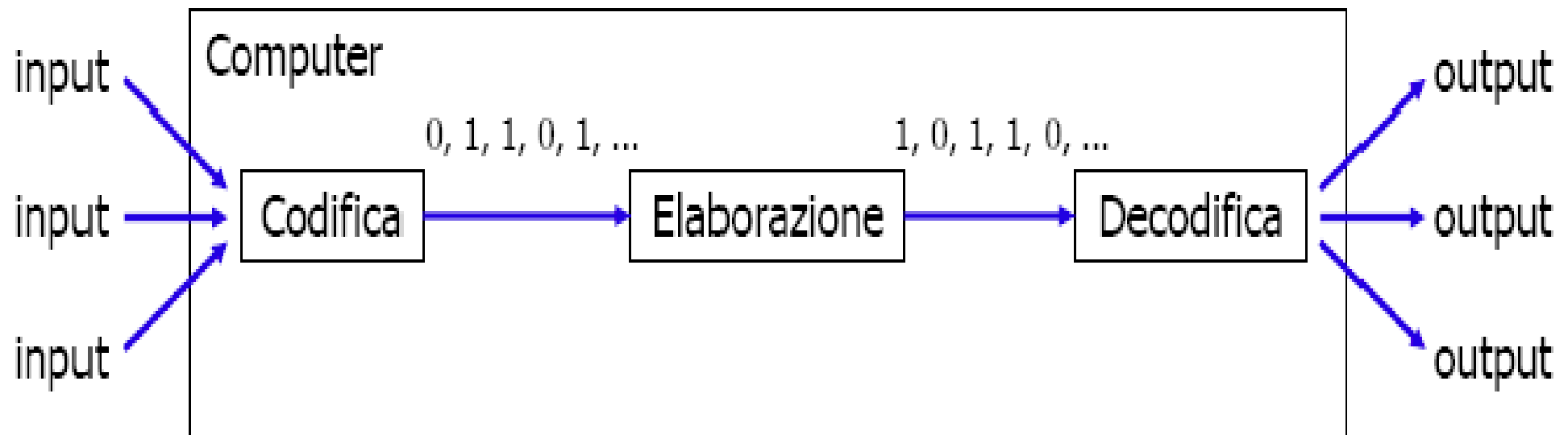
Perché solo due cifre?

Problemi di realizzazione

E più semplice ed affidabile costruire componenti hardware che siano in grado di “trovarsi” solo in due stati differenti.

Uno stato viene associato allo zero, uno stato viene associato all'uno, i computer possono memorizzare solo zeri e uni.

Rappresentazione ed elaborazione dell'informazione



Rappresentazione dell'Informazione

Come rappresentiamo
numeri interi,
parole,
immagini e suoni
con SOLO **DUE** cifre???

Cominciamo con un problema
diverso ma attinente...

Quanti oggetti diversi riesco a rappresentare/distinguere con...

....1 BIT?

dove rappresentare/distinguere vuol dire
**“associare ad ogni oggetto una sequenza
diversa di zeri e uno”**

Quanti oggetti diversi riesco a rappresentare/distinguere con...

....1 BIT?

Primo oggetto --> il bit a 
Secondo oggetto --> il bit a 
terzo oggetto --> ????

con 1 BIT distinguo tra $2=2^1$ oggetti diversi

Quanti oggetti diversi riesco a rappresentare/distinguere con...

....2 BIT?

Primo oggetto --> i bit a **00**
Secondo oggetto --> i bit a **01**
terzo oggetto --> i bit a **10**
quarto oggetto --> i bit a **11**
quinto oggetto --> ??????

con 2 BIT distinguo tra $4=2^2$ oggetti diversi

Quanti oggetti diversi riesco a rappresentare/distinguere con...

....3 BIT?

Primo oggetto --> i bit **000**
Secondo oggetto --> i bit a **001**
terzo oggetto --> i bit a **010**
quarto oggetto --> i bit a **011**
quinto oggetto --> **100**
sesto oggetto --> **101**
settimo oggetto --> **110**
ottavo oggetto --> **111**
nono oggetto --> ????

con 3 BIT distinguo tra $8=2^3$ oggetti diversi

Quanti oggetti diversi riesco a rappresentare/distinguere con...

- 1 BIT --> 2^1 oggetti diversi
- 2 BIT --> 2^2 oggetti diversi
- 3 BIT --> 2^3 oggetti diversi
- 4 BIT --> ?? oggetti diversi

Quanti oggetti diversi riesco a rappresentare/distinguere con...

- 1 BIT --> 2^1 oggetti diversi
- 2 BIT --> 2^2 oggetti diversi
- 3 BIT --> 2^3 oggetti diversi
- 4 BIT --> 2^4 oggetti diversi

Quanti oggetti diversi riesco a rappresentare/distinguere con...

- 1 BIT --> 2^1 oggetti diversi
- 2 BIT --> 2^2 oggetti diversi
- 3 BIT --> 2^3 oggetti diversi
- 4 BIT --> 2^4 oggetti diversi
-
- n BIT --> ?? oggetti diversi



Quanti oggetti diversi riesco a rappresentare/distinguere con...

- 1 BIT --> 2^1 oggetti diversi
- 2 BIT --> 2^2 oggetti diversi
- 3 BIT --> 2^3 oggetti diversi
- 4 BIT --> 2^4 oggetti diversi
-
- n BIT --> 2^n oggetti diversi

Numeri naturali

Come facciamo a rappresentare i

NUMERI NATURALI

in binario?

Sistema di numerazione posizionale in base

-La scelta della **base** stabilisce quante cifre si possono utilizzare per rappresentare i numeri:

- **BINARIO**: base 2, due cifre: 0,1
- **DECIMALE**: base 10, dieci cifre: 0,1,2,3,4,5,6,7,8,9
- **ESADECIMALE**: base 16, sedici cifre:
0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Sistema di numerazione posizionale in base

-La scelta della **base** stabilisce quante cifre si possono utilizzare per rappresentare i numeri:

- **BINARIO**: base 2, due cifre: 0,1

- **DECIMALE**: base 10, dieci cifre: 0,1,2,3,4,5,6,7,8,9

- **ESADECIMALE**: base 16, sedici cifre:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

-La **posizione** della cifra nel numero determina la sua

“importanza” (in 1.000.001, l'importanza dell'1 piu' a destra e' diversa da quello dell'1 piu' a sinistra...); la cifra piu' a destra e' la *meno significativa*, quella piu' a sinistra la *piu' significativa*.

Sistema di numerazione posizionale in base

-La scelta della **base** stabilisce quante cifre si possono utilizzare per rappresentare i numeri:

- **BINARIO**: base 2, due cifre: 0,1

- **DECIMALE**: base 10, dieci cifre: 0,1,2,3,4,5,6,7,8,9

- **ESADECIMALE**: base 16, sedici cifre:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

-La **posizione** della cifra nel numero determina la sua “importanza” (in 1.000.001, l'importanza dell'1 piu' a destra e' diversa da quello dell'1 piu' a sinistra...); la cifra piu' a destra e' la *meno significativa*, quella piu' a sinistra la *piu' significativa*.

-Esistono sistemi di numerazione non posizionale (esempio: i numeri romani)

Il testo

Viene associato un **codice binario** a
ciascun carattere

Quali caratteri?

Quale codice?

Quanti bit?

Il testo

CODICE ASCII

(American Standard Code for Information Interchange)

Originale: ogni carattere dell'alfabeto latino
(caratteri sia minuscoli che maiuscoli) +
punteggiatura + caratteri speciali + simboli
matematici....

viene codificato con 7 bit

Possibilità di codificare $2^7 = 128$ caratteri diversi

Il testo

CODICE ASCII ESTESO*

ad ogni carattere vengono associati 8 bit

Possibilità di codificare $2^8 = 256$ caratteri diversi

Ne esistono **varie versioni**:

- una per codificare i caratteri dei linguaggi dell'Europa occidentale (es. lettere accentate)
- una per i linguaggi dell'Europa orientale
- una per i caratteri cirillici

* spesso questo viene chiamato “codice ASCII”, mentre quello originale
US-ASCII

Il testo



Esempio di tabella di Codice ASCII (N.B. manca una colonna)

ATTENZIONE:
anche lo "spazio"
deve essere
codificato!



Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00100000	32	Spc	01000000	64	@	01100000	96	
00100001	33	!	01000001	65	A	01100001	97	a
00100010	34	"	01000010	66	B	01100010	98	b
00100011	35	#	01000011	67	C	01100011	99	c
00100100	36	\$	01000100	68	D	01100100	100	d
00100101	37	%	01000101	69	E	01100101	101	e
00100110	38	&	01000110	70	F	01100110	102	f
00100111	39	,	01000111	71	G	01100111	103	g
00101000	40	(01001000	72	H	01101000	104	h
00101001	41)	01001001	73	I	01101001	105	i
00101010	42	*	01001010	74	J	01101010	106	j
00101011	43	+	01001011	75	K	01101011	107	k
00101100	44	,	01001100	76	L	01101100	108	l
00101101	45	-	01001101	77	M	01101101	109	m
00101110	46	.	01001110	78	N	01101110	110	n
00101111	47	/	01001111	79	O	01101111	111	o
00110000	48	0	01010000	80	P	01110000	112	p
00110001	49	1	01010001	81	Q	01110001	113	q
00110010	50	2	01010010	82	R	01110010	114	r
00110011	51	3	01010011	83	S	01110011	115	s
00110100	52	4	01010100	84	T	01110100	116	t
00110101	53	5	01010101	85	U	01110101	117	u
00110110	54	6	01010110	86	V	01110110	118	v
00110111	55	7	01010111	87	W	01110111	119	w
00111000	56	8	01011000	88	X	01111000	120	x
00111001	57	9	01011001	89	Y	01111001	121	y
00111010	58	:	01011010	90	Z	01111010	122	z
00111011	59	;	01011011	91	[01111011	123	{
00111100	60	<	01011100	92	\	01111100	124	
00111101	61	=	01011101	93]	01111101	125	}
00111110	62	>	01011110	94	^	01111110	126	40
00111111	63	?	01011111	95	_	01111111	127	Del

Il testo

Come si usa il codice ASCII per rappresentare un testo in binario e viceversa?

Codifica: ad ogni carattere del testo si sostituisce la sequenza di bit corrispondente

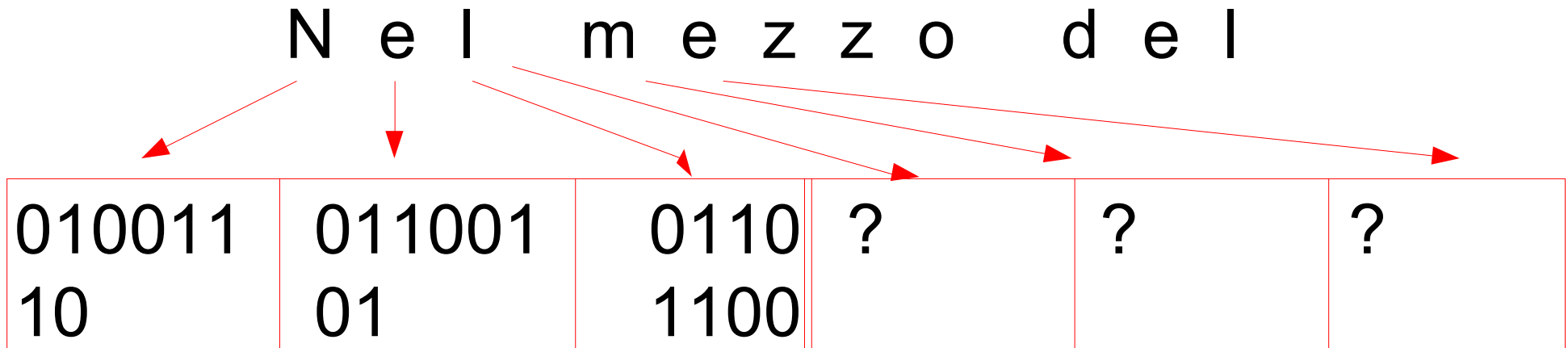
Decodifica: ad ogni sequenza di (7) 8 bit si sostituisce il carattere corrispondente

Il testo

Codifica: ad ogni carattere del testo si sostituisce la sequenza di bit corrispondente

ESEMPIO

N e l m e z z o d e l



010011	011001	0110	?	?	?
10	01	1100			

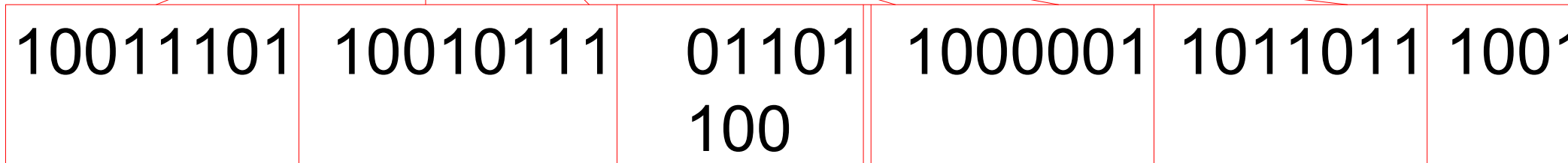
.....continua

Il testo

Decodifica: ad ogni sequenza di 7 bit si sostituisce il carattere corrispondente

ESEMPIO

N e l m e z z o d e l



.....continua

Il testo

Problema: cosa succede se la codifica e la decodifica vengono fatti usando versioni diverse del codice ASCII?



Il testo

Problema: cosa succede se la codifica e la decodifica vengono fatti usando versioni diverse del codice ASCII?

Risposta: il testo decodificato non e' identico al testo originale

Il testo

Problema: 256 combinazioni non bastano per codificare anche i linguaggi che non usano l'alfabeto latino

(es: cinese, giapponese, arabo, ebraico, alfabeto Braille, simboli matematici, chimici....)

SOLUZIONE: usare **16 bit** per la codifica di ogni carattere --> codice **UNICODE**
(codifica $2^{16} = 65.536$ caratteri diversi!)

Cos'e' il formato dei file?

I file vengono utilizzati per memorizzare dati e programmi nelle memorie *permanenti* dei computer....

... ma **per il computer un file non e' altro che una sequenza di bit...**

... e la sequenza e' stata creata in modo opportuno per poter rappresentare in binario i dati da memorizzare.

Per riuscire ad interpretare correttamente questa sequenza, il computer deve sapere **quale tipo di dato o programma e' stato codificato con quella sequenza di bit e con quale convenzione.**

Alcuni formati sono liberi, altri proprietari.

Cos'e' il formato dei file?

Il formato del file e' indicato dall'**estensione** nel nome del file.
L'estensione e' la parte finale del nome (generalmente 3 lettere)
che segue il punto.

L'estensione da' anche un'indicazione su quale programma ha
generato il file e quale e' in grado di aprirlo correttamente.

ATTENZIONE:

L'estensione del nome del file NON e' vincolante, puo' essere
cambiata senza cambiare il formato del contenuto

Formato dei File

File che contengono informazione di tipo **testuale**

File di testo semplice **.txt**

Prodotto da un **editore di testo**

codificato usando il codice ASCII o UNICODE
non contiene immagini

File di testo **.rtf** (rich text format)

Sviluppato dalla Microsoft

Letto e prodotto dalla maggior parte degli editori di testo

Mantiene la **formattazione** e le immagini del testo

File che contengono informazione di tipo testuale

File di testo **proprietario**

Prodotto da un word processor

Contiene informazioni relative alla **formattazione** del testo ed oggetti inseriti nel documento

Codificato usando codici proprietari

Es. .doc – Microsoft Word, .pages - Apple

Portable Document Format **.pdf**

Sviluppato da Adobe Systems

Rappresenta in documenti in maniera indipendente dal software e dal hardware che li genera

Formato aperto *: Chiunque puo' creare applicazioni che generano e leggono file pdf senza pagare diritti alla Adobe

MOLTO adatto alla **condivisione** dei documenti

(* quanti PDF-??? A-1 .. X ? https://it.wikipedia.org/wiki/Portable_Document_Format)



Analogico VS Digitale



Usiamo gli orologi per misurare il tempo

Dobbiamo trovare una codifica per rappresentare “**il tempo che passa**”, in modo da poterlo misurare con gli orologi

Gli orologi analogici ci danno una codifica “**continua**” del tempo che passa:

in teoria sono in grado di rappresentare ogni istante di tempo (anche se noi non siamo in grado di leggerlo)

Gli orologi digitali ci danno una codifica

“**discreta**” del tempo che passa:

non possiamo conoscere il dettaglio oltre quello previsto dall'orologio.

Dall'Analogico al Digitale

Codifica **ANALOGICA**

individuare una grandezza **analogica** tale che:
ad ogni incremento/decremento della prima corrisponda un
incremento/decremento della seconda

Codifica **DIGITALE**

introdurre un insieme (finito) di simboli e un insieme di
regole di codifica che specificano come ogni possibile
valore della grandezza sia rappresentato da una sequenza
dei simboli

DIGITALIZZARE

Rendere digitale; trascrivere valori analogici continui attraverso
segnali discreti: ad es. “digitalizzare un'immagine”

Immagini digitali



Come fanno a proiettare le immagini sullo schermo gigante?



Guardiamo più vicino....

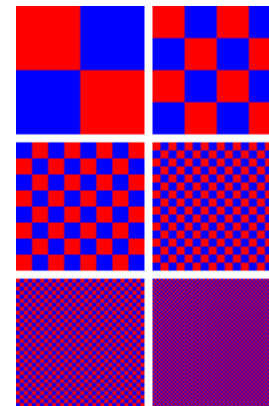


COSA SONO TUTTI QUEI PUNTINI COLORATI DIETRO LIGABUE????

Rappresentazione delle immagini

Come si rappresenta un'immagine in digitale?

Nel formato bitmap (o raster),
si usano un numero **FINITO** di elementi grafici minimi,
ognuno dei quali con un colore a scelta tra un insieme
FINITO di colori prestabiliti



Piu' e' piccolo l'elemento grafico minimo, piu' da lontano si guarda l'immagine,
meno si percepisce la scomposizione dell'immagine in elementi grafici

Immagine digitale

(bitmap o raster)

L'elemento grafico minimo si chiama
PIXEL

(pix <-- picture + el <-- element)

La **RISOLUZIONE** e' il numero di pixel che
compongono l'immagine
(dato come altezza x larghezza)

La **PROFONDITA'** di **COLORE** e' il numero di bit
che viene usato per codificare il colore di ogni
singolo pixel

Immagine digitale (bitmap) CODIFICA

Dati il numero di pixel e la profondità di colore,
riga per riga
(partendo dall'alto verso il basso)
e per ogni riga da sinistra a destra,
si memorizza il colore di ogni pixel

Esempio

Risoluzione 8x8 Profondità di colore 24 bit (RGB)
(16.777.216 colori)

```
0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
00FF00 FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF 00FF00
00FF00 FFFFFFFF FF0000 FFFFFFFF FF0000 FFFFFFFF FF0000 00FF00
0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF 0000FF
```

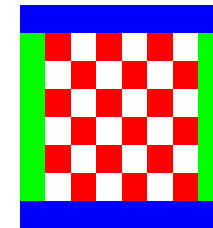
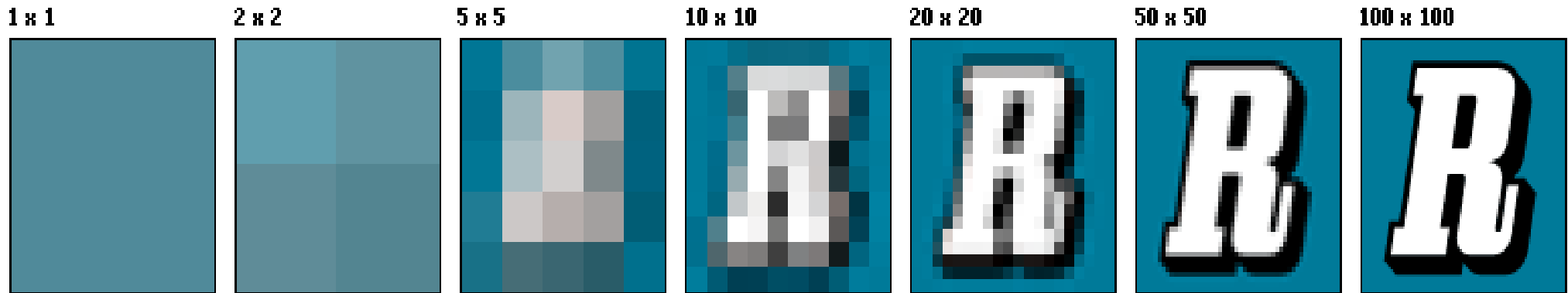


Immagine digitale (bitmap)

RISOLUZIONE
(numero di pixel)

aumentando il numero di pixel
aumenta la nitidezza dell'immagine



Se si ingrandisce troppo un'immagine digitale,
l'immagine può perdere nitidezza e i pixel
possono diventare apprezzabili ad occhio nudo

Immagine digitale (bitmap)

PROFONDITA' di COLORE

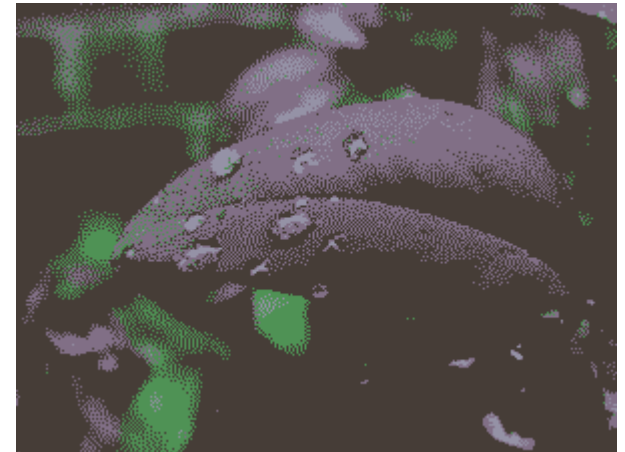
(numero di bit per pixel)

ogni bit aggiunto raddoppia il numero di colori disponibili

Esempio



1 bit
(2 colori)



2 bit
(4 colori)



4 bit
(16 colori)



8 bit
(256 colori)

Rappresentazione delle immagini

Tecnica BITMAP

PROBLEMA: i **file** generati con questa tecnica sono spesso troppo **grandi**

Sono stati sviluppati schemi di **compressione**

Permettono di ridurre il numero di bit usati per memorizzare l'immagine

Compressione

Cosa vuol dire **comprimere** un file?

Cambiare la modalita' in cui abbiamo **codificato** le informazioni in modo da utilizzare meno bit per la sua memorizzazione (proprieta' utile anche per la trasmissione)

Compressione **lossless**: la nuova codifica **non provoca perdita d'informazione** (indispensabile in alcuni casi: file eseguibili)

Compressione **lossy**: la nuova codifica **provoca perdita d'informazione** (accettabile in alcuni casi: immagini, audio), ma garantisce comunque l'intelligibilita' delle informazioni nel file compresso

Compressione

Esempio – compressione

sequenza di 458 bit

1111111.....1111....1111...1111....0000...
0000.....000....00000....0000111....1.....
...1111..111111

Utilizzo MOLTO meno spazio se descrivo la sequenza di bit nel modo seguente

253 uno seguiti da 118 zero seguiti da 87 uno

RLE Run Length Encoding

e, in questo caso, non perdo informazione

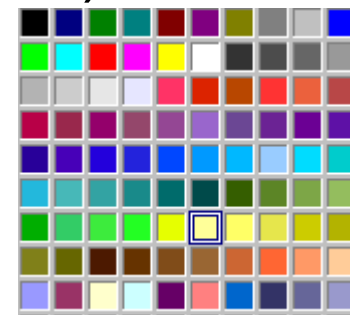
Rappresentazione delle immagini – formati compressi

GIF: Graphical Interchange Format

Supporta **256 colori** a immagine.

I colori sono descritti basandosi sul modello RGB
(che supporta circa 16 milioni di colori
utilizzando 24 bit per colore)

Costruisce una tabella di colori (**palette** o tavolozza)
associando ad ogni
colore solo 8 bit



Il sistema GIF e' **senza perdita**
(si applicato solo ad immagini con 256 colori)

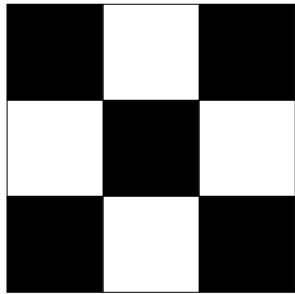
OK per videogiochi, Web – **KO** per fotografia professionale

Rappresentazione delle immagini – formati compressi

JPG: Joint Photographic Experts Group, LOSSY

Si trae vantaggio dalle **limitazioni dell'occhio** umano:
e' piu' sensibile ai cambiamenti di luminosita' che di colore

Se i quadrati sono abbastanza piccoli....



.... una
scacchiera...

...viene
percepita...



...come un
quadrato grigio

Memorizzare un grande
quadrato grigio, richiede
meno spazio di una
grande scacchiera
bianco/nero

Esempio: la qualita' varia all'aumentare del
grado di compressione, da sinistra verso destra



E' diventato lo standard adottato dai produttori di macchine fotografiche digitali

Ricerca per immagini

Come fanno i motori di ricerca a trovare immagini simili a quella che indichiamo noi?



Dimensioni immagine:
2592 × 1458

Impossibile trovare questa immagine con altre dimensioni.

[Immagini visivamente simili](#) - Segnala immagini non appropriate



Si basano sul colore dei pixel e sulla loro disposizione nelle immagini.

IDEA di fondo:

Se due immagini hanno tanti pixel dello stesso colore nelle stesse posizioni... forse l'immagine che compongono e' simile!