



UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA

DIMES

Codifica binaria dell'informazione

Codifica binaria
Elaborazione di dati binari

I U / e

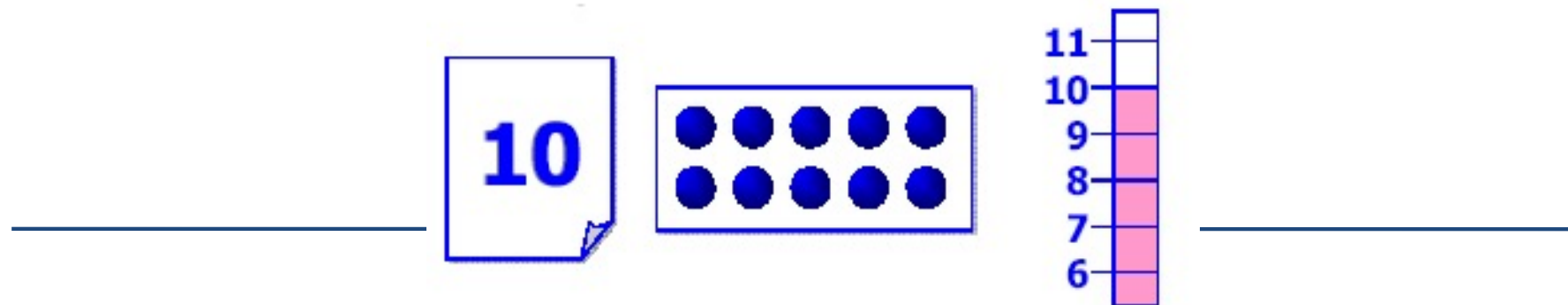
Technische Universiteit
Eindhoven
University of Technology

Codifica dell'informazione

- L'informazione gestita dai sistemi di elaborazione deve essere **codificata**
 - per poter essere memorizzata, elaborata, scambiata,...
- ~~Necessario codificare dati e istruzioni~~
 - Algoritmo = insieme di istruzioni che operano su dati
 - Per eseguire un programma è necessario codificare e memorizzare sia i dati sia il programma (le istruzioni)
- L'esecutore automatico deve essere in grado di:
 - Memorizzare istruzioni e dati
 - Manipolare istruzioni e dati

Informazione e codifica

- La stessa informazione si può codificare in modi differenti



- Stessa codifica per informazioni differenti



Sistemi di codifica

- Detto anche “codifica” o “codice”
 - Usa un insieme di simboli di base (**alfabeto**)
 - I simboli dell’alfabeto possono essere combinati ottenendo differenti **configurazioni**
-
- Sono distinguibili l’una dall’altra
 - Sono dette anche “codici”, “stati”
- Associa ogni configurazione ad una particolare entità di informazione
 - la configurazione diventa un modo per rappresentarla

Sistemi di codifica: numeri (es.)

- **Alfabeto**
 - cifre: “0”, “1”, “2”, ..., “9”
 - separatori: decimale (“,”), migliaia (“.”)
 - segni: positivo (“+”), negativo (“-”)

- **Regole di composizione (sintassi)**
 - definiscono le combinazioni ammissibili (ben formate)
 - 12.318,43: OK
 - 12,318,43: **ERRORE!**

- **Codice (semantica)**
 - Associano ad ogni configurazione un’entità di informazione
 - $2.318,43 = 2 \times 10^3 + 3 \times 10^2 + 1 \times 10^1 + 1 \times 10^0 + 4 \times 10^{-1} + 3 \times 10^{-2}$

- **Sistemi diversi possono usare lo stesso alfabeto**
 - $123,456 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3}$ [IT]
 - $123,456 = 1 \times 10^5 + 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$ [UK]

Sistemi di codifica

- Vedremo brevemente sistemi di codifica per:
 - Codifica di numeri
 - Codifica di caratteri
 - Codifica di immagini

 - Codifica di sequenze sonore
- Esistono standard internazionali per risolvere problemi di compatibilità
 - tra differenti sistemi software di tipo e marca diversi

Codifica binaria

- Codifica binaria: usa un alfabeto di **2** simboli
- Usata nei sistemi informatici (livello fisico)
 - si usa una grandezza fisica (luminosità, tensione elettrica, la corrente elettrica) per rappresentare informazione

 - si divide in intervalli l'insieme dei valori che la grandezza può assumere: ogni intervallo corrisponde ad un simbolo
- Solo 2 simboli per ridurre la probabilità di errore
 - tanti più simboli si devono distinguere e tanto meno la rivelazione sarà affidabile in presenza di “rumore”



Codifica binaria

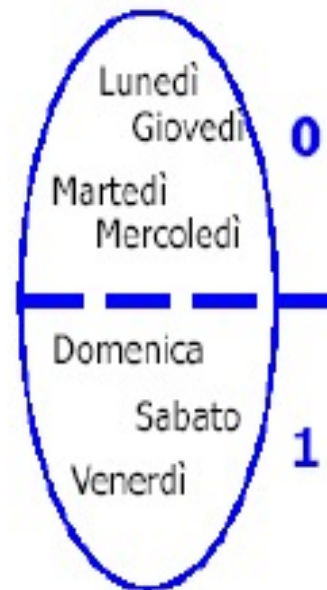
- **BIT (Binary digit)**
 - unità elementare di informazione rappresentabile con dispositivi elettronici
 - con 1 bit si possono rappresentare 2 stati
 - 0/1, on/off, si/no

- Combinando più bit si può codificare un numero maggiore di stati
 - con 2 bit possono rappresentare 4 stati
 - con **K** bit si possono rappresentare **2^K** stati
- Quanti bit servono per codificare N stati?
 - $N \leq 2^K \rightarrow K \geq \log_2 N \rightarrow \mathbf{K = \lceil \log_2 N \rceil}$

Codifica binaria: unità di misura

- **BYTE** = 8 bit
 - può rappresentare $2^8 = 256$ stati
- **KiloByte** (KB) = 2^{10} byte = 1024 byte $\cong 10^3$ byte
- ~~**MegaByte** (MB) = 2^{20} byte $\cong 10^6$ byte~~
- **GigaByte** (GB) = 2^{30} byte $\cong 10^9$ byte
- **TeraByte** (TB) = 2^{40} byte $\cong 10^{12}$ byte

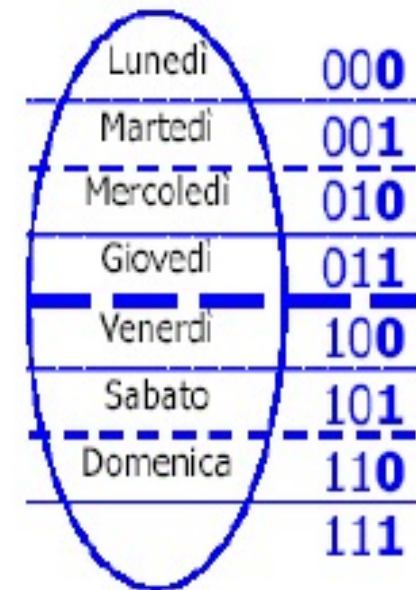
Es.: i giorni della settimana in binario



1 bit
2 "gruppi"



2 bit
4 "gruppi"



3 bit
8 "gruppi"

Codifica dei caratteri

Codice ASCII (standard)

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del

LA CALABRIA

UNIVERSITÀ
 DI
 CROTONE
 DIPARTIMENTO DI
 INFORMATICA,
 TELECOMUNICAZIONI
 E ELETTRONICA

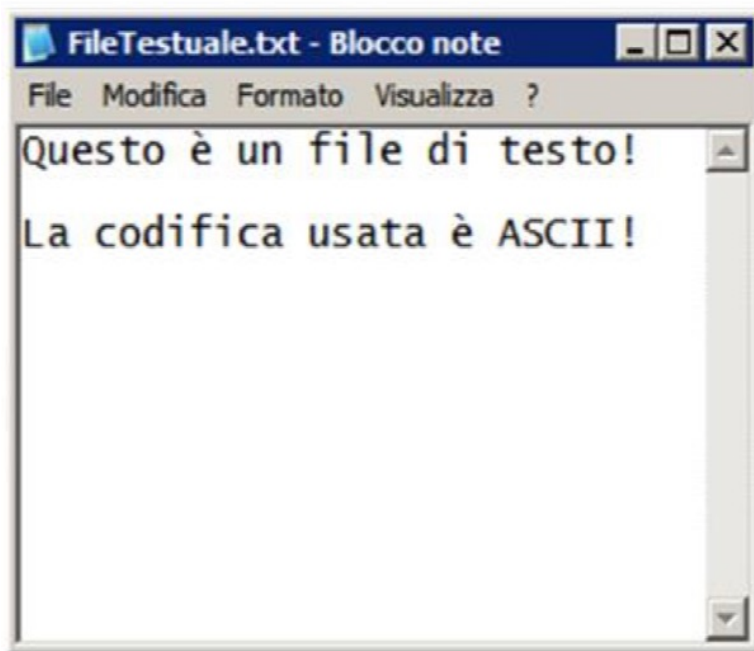
CODIFICA TESTO

testo visualizzato

txt

dec

bin



Ques	081117101115	01010001011101010110010101110011
to è	116111032138	01110100011011110010000010001010
un	032117110032	00100000011101010110111000100000
file	102105108101	01100110011010010110110001100101
di	032100105032	00100000011001000110100100100000
test	116101115116	01110100011001010111001101110100
o!	111033013010	01101111001000010000110100001010
La	013010076097	00001101000010100100110001100001
cod	032099111100	00100000011000110110111101100100
ific	105102105099	01101001011001100110100101100011
a us	097032117115	01100001001000000111010101110011
ata	097116097032	01100001011101000110000100100000
è AS	138032065083	10001010001000000100000101010011
CII!	067073073033	01000011010010010100100100100001

Codifica di caratteri

- Si associa un codice ad ogni simbolo dell'alfabeto
- Codifica **ASCII**
 - Caratteri speciali, punteggiatura, a-z, A-Z, 0-9
 - Utilizza 7 bit (128 caratteri)
 - I codici ASCII estesi usano 8 bit (256 caratteri)
- Codifica **EBCDIC**
 - Utilizza 8 bit (256 caratteri)
- Codifica **UNICODE**
 - Utilizza 16 bit (65536 caratteri)
 - I primi 128 caratteri sono gli stessi di ASCII
 - Gli altri corrispondono ad altri alfabeti (greco, cirillico,...)
 - Non copre i simboli (oltre 200.000) di tutte le lingue!

Codifica di immagini



```
001011111100001010101000000000010101  
000010000000000000000000000000000111  
010100000001110010010100010001010111  
000100010001000101011101000100010000  
100001010101000111100011101100100100  
101010100000101010000100100010001001  
01010100101010101000000000010101000010  
00000000000000000000000000000111010100  
000001110010010100010001010111000100  
010001000101011101000100010000100001  
010101000111100011101100100100101000  
11110001
```

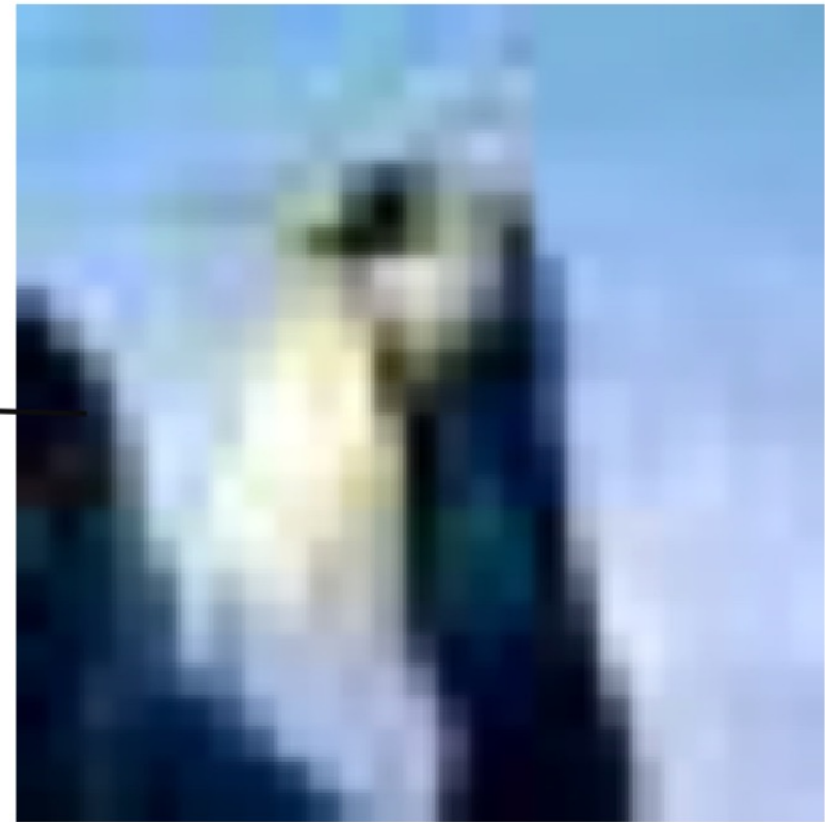
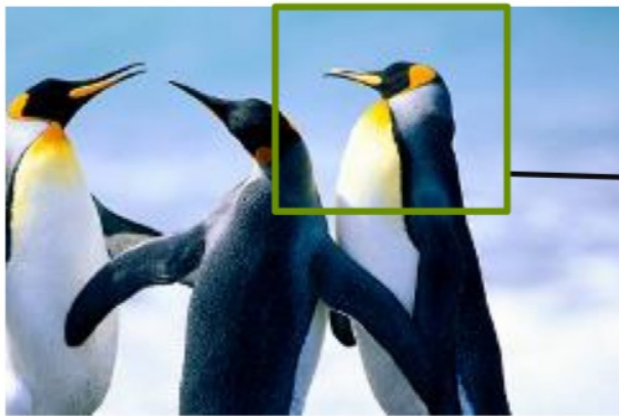


Immagini Digitali

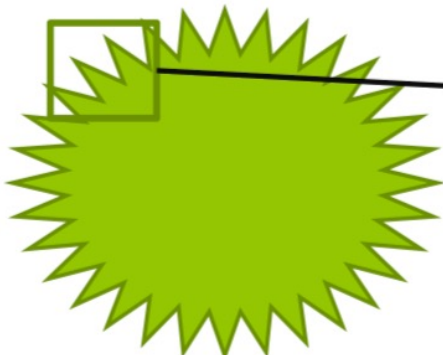
Tipo	Definizione	Proprietà
Raster o Bitmap	Mappata all'interno di una griglia, come un grande mosaico. La grandezza della griglia dipende dalla risoluzione dell'immagine.	Rappresentazione più semplice (richiesta poca elaborazione) Spazio maggiore per essere memorizzate.
Vettoriale	Basate su forme e colori generate tramite formule matematiche	Ingrandimento teoricamente infinito. Rappresentazione più complessa

Immagini Digitali: Ingrandimento

Raster ○ **Bitmap**

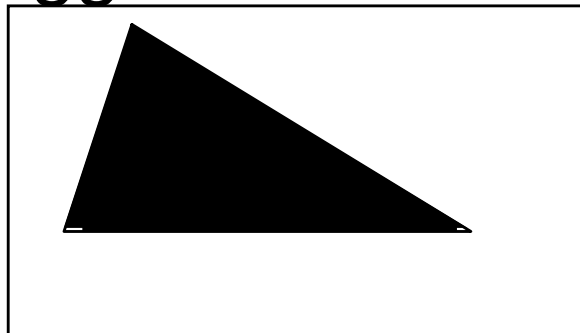


Vettoriale

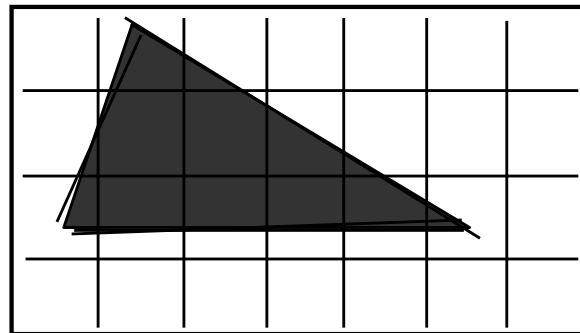


Codifica di immagini

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



- Si suddivide l'immagine con una griglia formata da righe orizzontali e verticali a distanza costante

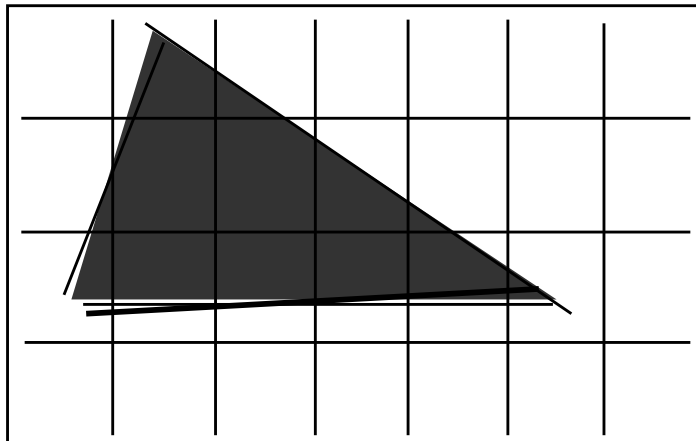


Codifica di immagini

- **pixel** (picture element)
 - ogni quadratino derivante dalla suddivisione dell'immagine
- Codifica di un pixel:
 - il simbolo “**0**” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino bianco (in cui il bianco è predominante)
 - il simbolo “**1**” viene utilizzato per la codifica di un pixel corrispondente ad un quadratino nero (in cui il nero è predominante)

Codifica di immagini

- Poiché una sequenza di bit è lineare, si deve definire una convenzione per ordinare i pixel della griglia
 - Hp: assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



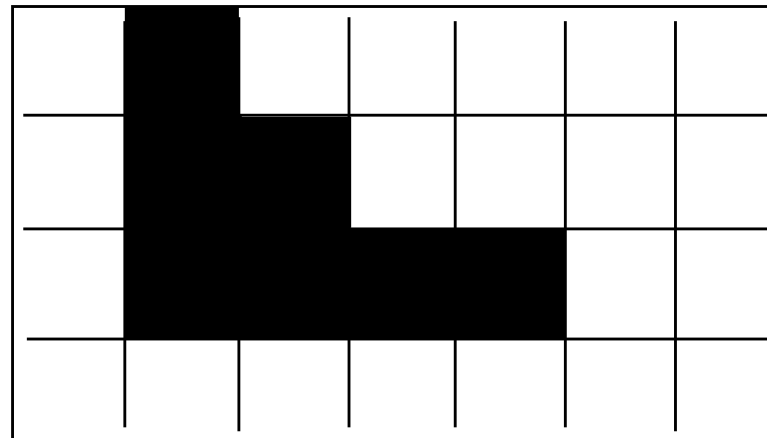
0 ₂₂	1 ₂₃	0 ₂₄	0 ₂₅	0 ₂₆	0 ₂₇	0 ₂₈
0 ₁₅	1 ₁₆	1 ₁₇	0 ₁₈	0 ₁₉	0 ₂₀	0 ₂₁
0 ₈	1 ₉	1 ₁₀	1 ₁₁	1 ₁₂	0 ₁₃	0 ₁₄
0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇

- La rappresentazione della figura è data dalla stringa:

000000 0111100 0110000 0100000

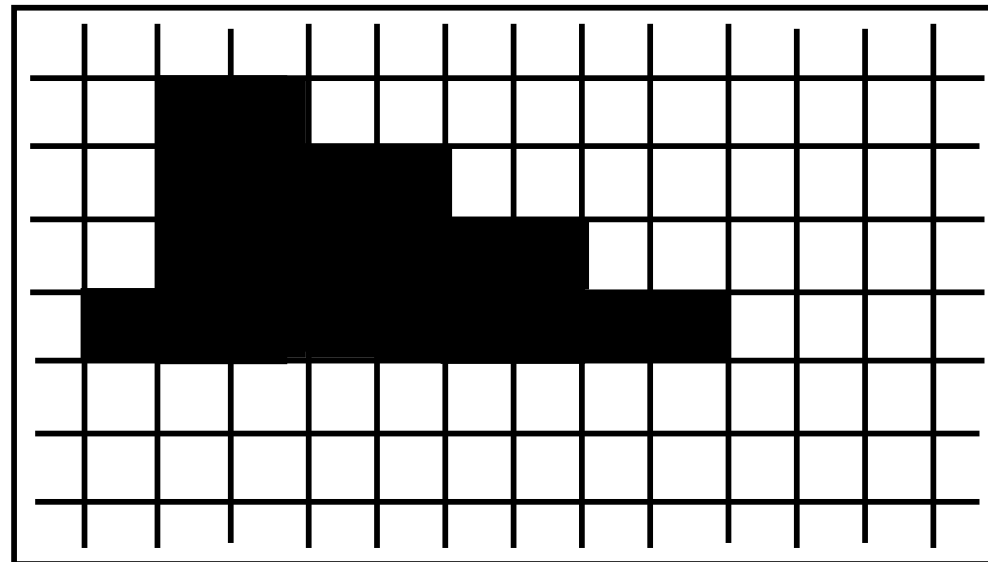
Codifica di immagini

- Approssimazione:
 - nella codifica si ottiene un'approssimazione della figura originaria
 - non sempre il contorno della figura coincide con le linee della griglia
 - Riconvertendo in immagine la stringa **0000000011110001100000100000** si ottiene:



Codifica di immagini





- La rappresentazione sarà più fedele all'aumentare del numero di pixel
 - ossia al diminuire delle dimensioni dei quadratini della griglia in cui è suddivisa l'immagine



Immagini Digitali: Bitmap

Immagini B/N e a gradazioni di grigio

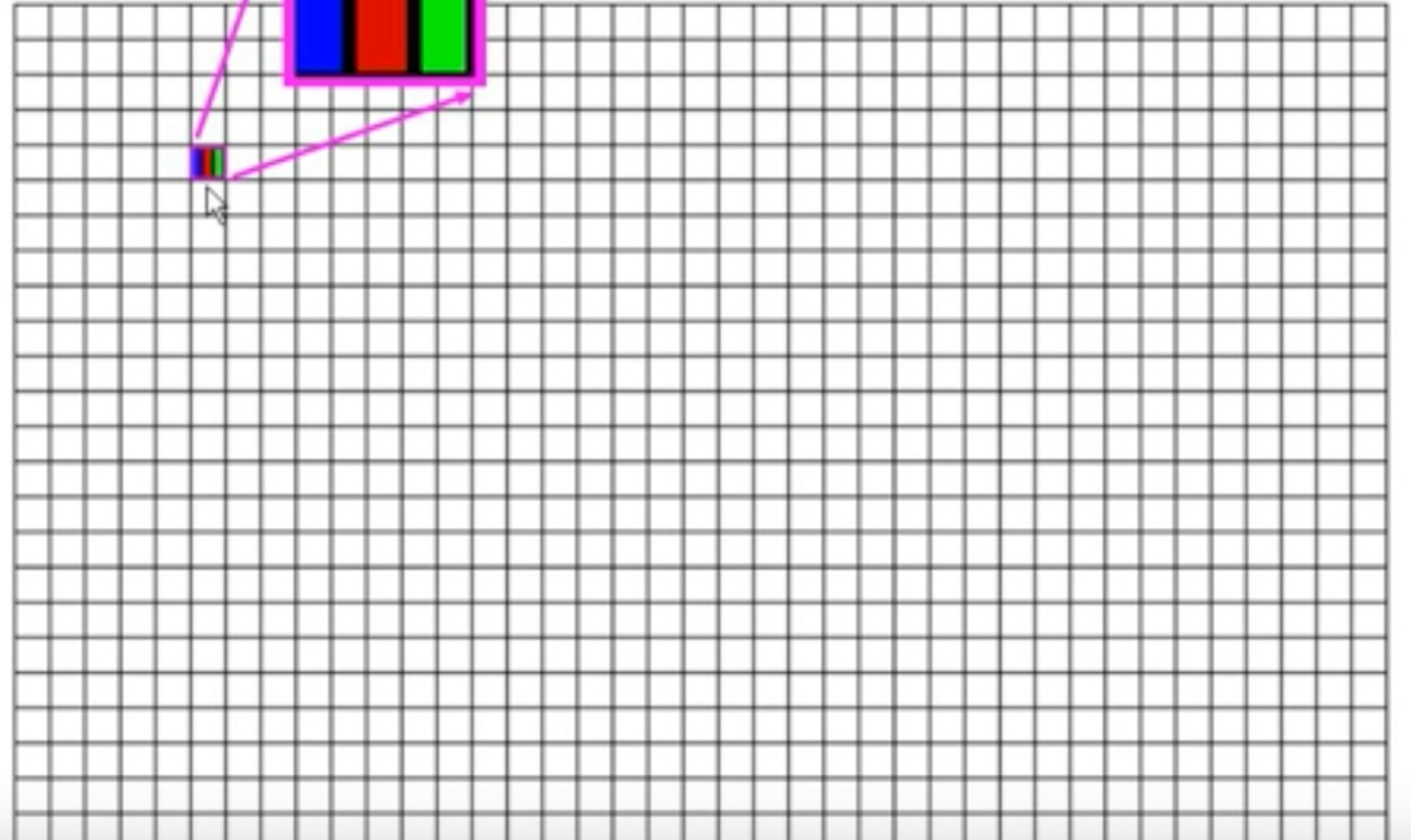
Esempio: con 4 bit possiamo rappresentare $2^2 = 4$ colori diversi

-  00
-  11
-  10
-  01



01	01	10	11
01	10	11	00
10	11	00	00
11	00	00	00

Pixel



Immagini con toni di grigio

- Le consuete immagini “in bianco e nero” hanno delle sfumature (**livelli di intensità di grigio**)
- Per codificare immagini con sfumature:

- Per memorizzare un pixel non basta un solo bit

Immagini Digitali: Bitmap

Immagini a Colori: codifica RGB

CODIFICA RGB (Red, Green, Blue): Qualsiasi colore può essere rappresentato dalla composizione del **Rosso**, del **Verde** e del **Blu** (colori primari).

Ogni pixel viene rappresentato con una combinazione dei tre colori. Rappresentando ogni colore primario con N°8 Bit (1 Byte) allora
3 Byte per ogni Pixel.

NOTA: con 8 bit per ogni colore primario rappresentiamo
 $2^8 = 256$ sfumature per ogni colore

E quindi, con la codifica **RGB** riusciamo a rappresentare
 $256 \times 256 \times 256 = 16.777.216$ colori diversi

ESEMPIO: Se un pixel deve essere di colore  la sua codifica **RGB** è:

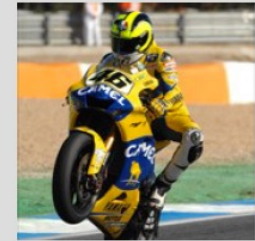
11100101 11100101 11100101

- Esempi:

R (rosso)	G (verde)	B (blu)	Colore
0	0	0	Nero
0	0	128	Blu pieno
128	128	128	Grigio 50%
255	0	0	Rosso pieno
255	255	0	Giallo pieno
153	51	102	Bordeaux
53	94	0	Verde oliva
255	255	255	Bianco
102	51	0	Marrone
255	204	153	Pesca

Occupazione di memoria

- 170x170 pixel, RGB a 8 bit per componente = 693.600 bit = **86.7 Kb**
- 32x32 pixel, RGB a 8 bit per componente = 24.576 bit = **3 Kb**
- 170x170 pixel, RGB a 3 bit per componente = 260.100 bit = **32.5 Kb**
- 16x16 pixel, RGB a 2 bit per componente = 512 bit = **64 byte**



Immagini a colori

UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA

DIMES

- Il numero di byte richiesti dipende da
 - **risoluzione**
 - **numero di colori** che ogni pixel può assumere
- I monitor tipici utilizzano
 - risoluzione: 640×480, 1024×768, 1280×1024
 - numero di colori per pixel: da 256 fino a 16 milioni
- Tecniche di **compressione**
 - riducono notevolmente lo spazio occupato dalle immagini

Immagini Digitali: Bitmap

Tecniche di Compressione

Esistono tecniche di compressione delle informazioni che consentono di ridurre drasticamente lo spazio occupato dalle Immagini

codifiche di compressione: le più famose sono

- la **CompuServe Graphic Interface (GIF)** (estensione **.gif**)
- **Joint Photographic Experts Group (JPEG)** (estensione **.jpg** o **.jpeg**)

Usano un sistema per **comprimere** l'informazione prima di memorizzarla e per **decomprimerla** prima di visualizzarla.

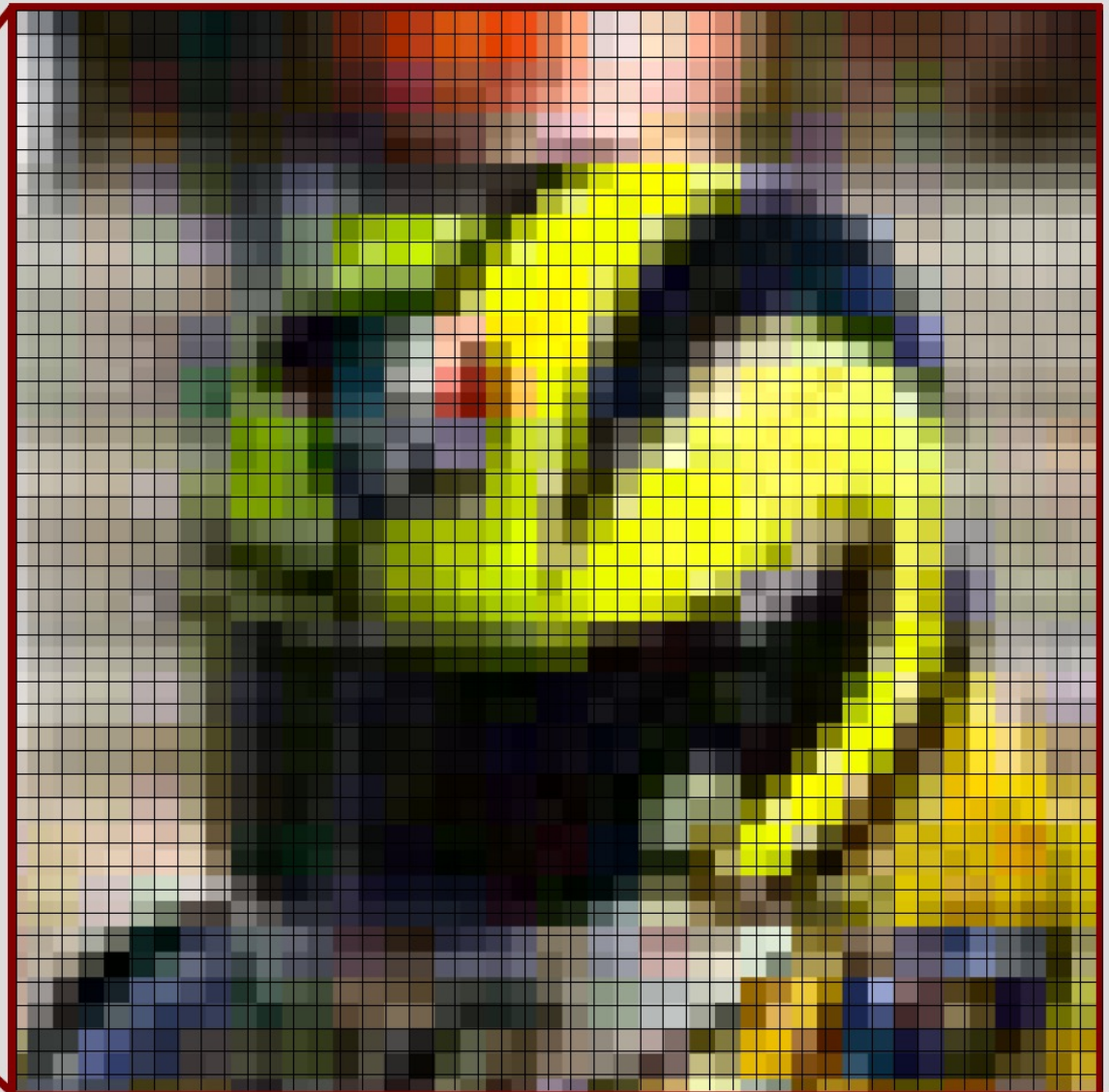
- tendono ad eliminare i pixel ripetitivi,
- sono compressioni con **perdita di informazione**. Tale perdita non può essere recuperata in alcun modo. La codifica JPEG consente di manipolare tale fattore di compressione.

Immagini Digitali: Bitmap

Estensioni

- jpeg
- gif
- png
- tga
- tiff
- raw
- bmp

IMMAGINI RASTER



Immagini raster

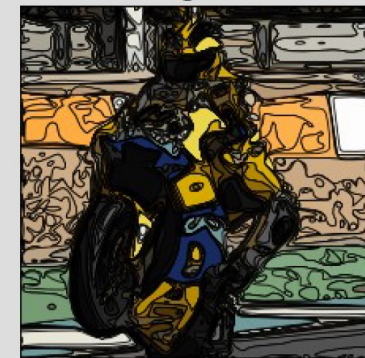
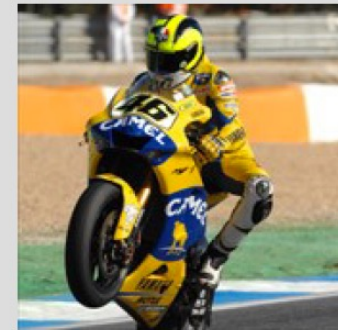


UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA

DIMES

- Distinguiamo subito due tipi di immagini
 - Le immagini **raster** sono di tipo fotografico; si rappresenta individualmente ogni singolo punto dell'immagine
 - Le immagini **vettoriali** sono essenzialmente disegni; si rappresentano i punti e le curve che formano il disegno



Immagini Digitali: Vettoriali

Come si memorizzano?

Una immagine vettoriale è descritta come un insieme di primitive geometriche alle quali possono essere attribuiti colori e anche sfumature.

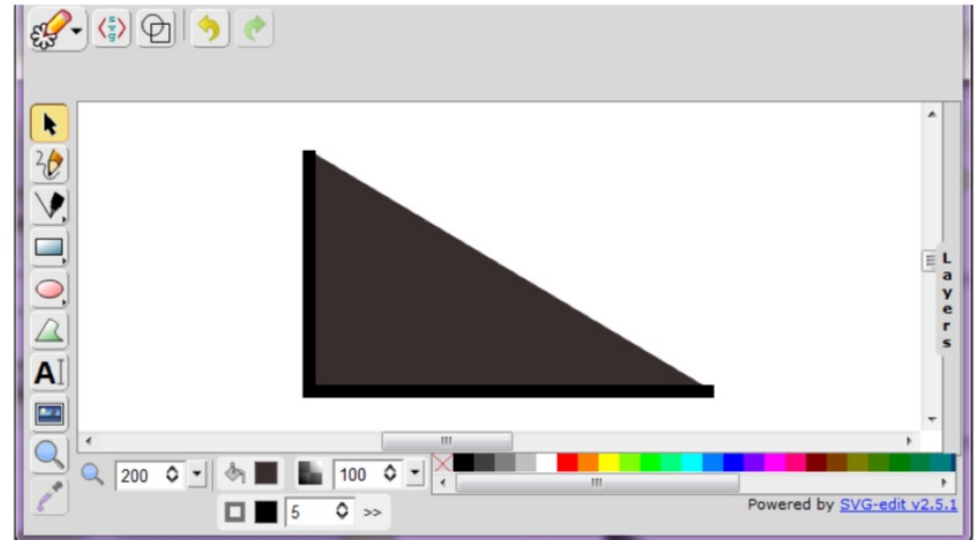
PRIMITIVE GEOMETRICHE

- Punti
- Linee
- Segmenti
- Triangoli

Immagini Digitali: Vettoriali Come si memorizzano?



Creazione



```
<svg width="640" height="480" xmlns="http://www.w3.org/2000/svg">
  <!-- Created with SVG-edit - http://svg-edit.googlecode.com/ -->
  <g>
    <title>Livello 1</title>
    <path id="svg_15" d="m125,169l0,93l156,0" stroke-linecap="null" stroke-linejoin="null" stroke-
dasharray="null" stroke-width="5" stroke="#000000" fill="#382e2e"/>
  </g>
</svg>
```

Immagini Digitali: Vettoriali

Vantaggi

1. Immagine espressa in una forma *'direttamente'* comprensibile dall'uomo e quindi direttamente modificabile (come per i formati SVG)



```
<svg width="640" height="480" xmlns="http://www.w3.org/2000/svg">
<!-- Created with SVG-edit - http://svg-edit.googlecode.com/ -->
<g>
<title>Livello 1</title>
<ellipse ry="63" rx="102" id="svg_1" cy="147" cx="176" stroke-width="5" stroke="#000000" fill="#FF0000"/>
<ellipse ry="11" rx="15" id="svg_2" cy="123" cx="133" stroke-width="5" stroke="#000000" fill="#332121"/>
<ellipse ry="11.5" rx="13.5" id="svg_3" cy="119.5" cx="206.5" stroke-width="5" stroke="#000000" fill="#332121"/>
<path d="m117,159c0,0 2,1 2,3c0,2 0.61732,2.07613 1,3c0.5412,1.30656 1,2 2,2c1,0 0.69344,1.4588 2,2c0.92388,0.38269
2,0 3,0c1,0 2.07613,0.61731 3,1c2.61313,1.0824 2.07613,2.61731 3,3c1.30656,0.5412 2.85274,3.1731 4,4c1.814,1.30745
1.82375,2.48625 4,3c0.97325,0.22975 3,0 4,0c2,0 2.09789,0.82443 4,2c1.70131,1.05147 3,0 4,0c2,0 2,1 3,1c1,0 2.03874,-
0.48055 5,0c3.12144,0.50655 6,2 8,2c2,0 3,0 5,0c3,0 6,0 7,0c1,0 2,0 3,0c2,0 6,0 9,0c1,0 4,0 5,0c1,0 3.01291,0.16019 0.51375
-4.88152,-3.19028 -8,-5c-1.93399,-1.12234 -2.38687,-1.9176 -5,-3c-1.84776,-0.76537 -5.11832,-4.52814 -6,-5c-3.17892,-
1.70131 ecap="null" stroke-linejoin="null" stroke-dasharray="null" stroke-width="5" stroke="#000000" fill="none"/>
</g>
</svg>
```

Immagini Digitali: Vettoriali

Vantaggi

2. Spazio per memorizzazione immagine molto più piccolo rispetto ad un equivalente raster.



RASTER	VETTORIALE
11.084 byte	2.133 byte

Immagine Digitali: Vettoriali

Vantaggi

3. Ingrandimento potenzialmente infinito senza perdita di qualità.



Immagini Digitali: Vettoriali

Estensioni

- ai (Adobe Illustrator)
- cdr (Corel Draw)
- svg (adatta per il WEB)**
- drv (Micrografx Designer/Draw file)
- dgn (Microstation)
- dxf (Drawing Interchange (eXchange))
- edrw (eDrawing)
- flt (OpenFlight format - Creator)
- fla (Macromedia Flash)
- igs (file di scambio CAD)
- lfp (Laser file plus)
- par (file parametrico)
- prt (Vari)
- sat (Acis 2D/3D Grafica vettoriale)
- pln (Archicad)
- pdf (formato Adobe riconosce bitmap e vettoriale)
- eps (riconosce sia bitmap che vettoriale)

Compressione JPEG (esempio)



UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI
INGEGNERIA INFORMATICA,
MODELLISTICA, ELETTRONICA
E SISTEMISTICA

Codifica Bitmap

- 800 x 600

- 16,8 mln colori (24 bit)

dimensione = 1.440.000 byte

≈ 1406 KB

Fattore qualità 90% (253 KB)



Fattore qualità 10% (12 KB)



Fattore qualità 1% (9 KB)

