

RAPPRESENTAZIONE DELLE INFORMAZIONI

RAPPRESENTAZIONE DELLE INFORMAZIONI

- Le informazioni gestite dai sistemi di elaborazione devono essere **codificate**
 - per poter essere memorizzate, elaborate, scambiate, ...
- In un elaboratore bisogna codificare sia i dati che le istruzioni
- I **sistemi di codifica** definiscono un insieme di simboli, e delle regole per combinare i simboli

RAPPRESENTAZIONE DELLE INFORMAZIONI

- Necessità di standard internazionali per risolvere i problemi di compatibilità
 - tra differenti sistemi software
 - tra calcolatori di tipo e marca diversi
- Cosa vedremo:
 - Codifica binaria
 - Codifica dei numeri naturali
 - Codifica dei numeri interi e razionali
 - Codifica dei caratteri
 - Codifica dei dati multimediali

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, vero/falso
- 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 sono necessari per codificare N oggetti?
 - $N \leq 2^K \rightarrow K \geq \log_2 N \rightarrow \mathbf{K = \lceil \log_2 N \rceil}$

CODIFICA BINARIA: UNITA' DERIVATE

- **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

CODIFICA DEI NUMERI NATURALI

- Sistema di numerazione posizionale con **base β**
 - β simboli (**cifre**) corrispondono ai numeri da 0 a $\beta-1$
 - i numeri naturali maggiori o uguali a β possono essere rappresentati da una sequenza di cifre
- Se un numero naturale N è rappresentato in base β dalla sequenza di n cifre

$$a_{n-1} a_{n-2} \dots a_1 a_0$$

allora N può essere espresso come segue:

$$N = \sum_{i=0}^{n-1} a_i \beta^i = a_{n-1} \beta^{n-1} + a_{n-2} \beta^{n-2} + \dots + a_2 \beta^2 + a_1 \beta + a_0$$

CODIFICA DEI NUMERI NATURALI

Esempio: 13 può essere espresso in funzione delle potenze di 2 come:

$$13 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$13 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

Cioè può essere rappresentato dalla sequenza di bit

1 1 0 1

CODIFICA DEI NUMERI INTERI

- Si utilizzano codifiche quali
 - **Bit di segno**
 - Il bit più a sinistra rappresenta il segno del numero (0 = '+', 1 = '-')
 - **Complemento a 1**
 - **Complemento a 2**

CODIFICA DEI NUMERI RAZIONALI

- **Fixed point (virgola fissa)**

- Un numero razionale è rappresentato come una coppia di numeri interi: la parte intera e la parte decimale
- *Es:* $12,52 \rightarrow \langle 12; 52 \rangle$

- **Floating point (virgola mobile)**

- Un numero razionale è rappresentato come un intero moltiplicato per una opportuna potenza di 10, cioè con una coppia <mantissa, esponente>
- *Es:* $12,52 = 1252/100 = 1252 * 10^{-2} \rightarrow \langle 1252; -2 \rangle$

CODIFICA DEI CARATTERI

- Codifica **ASCII**: associando un simbolo dell'alfabeto ad ogni numero possiamo codificare tutte le lettere
 - a-z A-Z 0-9usando 8 bit (cioè in un byte)!!
- Esempio: **00000101** rappresenta la lettera 'c'

OPERAZIONI LOGICHE

- **Valori di verità** (booleani):
 - **FALSO** = 0
 - **VERO** = 1
- **Operatori logici**
 - **X AND Y** = VERO se X = VERO e Y = VERO
 - **X OR Y** = VERO se X = VERO oppure Y = VERO
 - **NOT X** = VERO se X = FALSO
- Gli operatori logici permettono di esprimere operazioni su bit

CODIFICA DI DATI MULTIMEDIALI

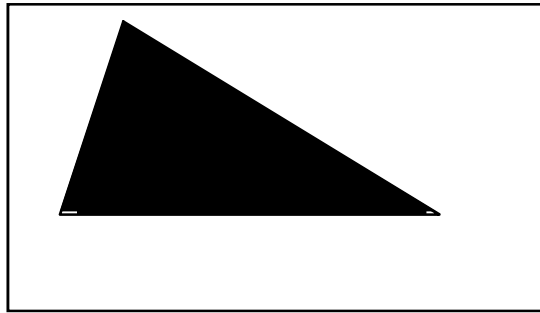
- Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori ma si diffondono sempre di più applicazioni che usano ed elaborano anche altri tipi di informazione:
 - diagrammi
 - immagini
 - suoni
- Spesso in questi casi si parla di applicazioni di tipo **multimediale**

CODIFICA DI IMMAGINI

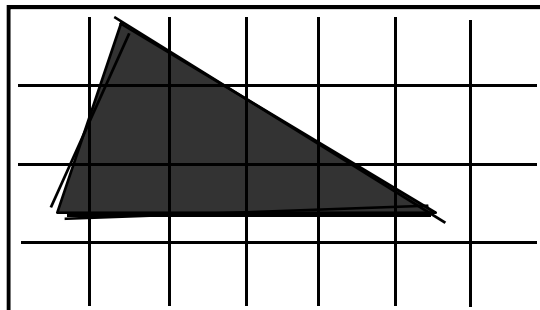
- Esistono numerose tecniche per la memorizzazione digitale e l'elaborazione di un'immagine
- Immagini = sequenze di bit!
- L'immagine viene digitalizzata cioè rappresentata con sequenze di pixel
- Ogni pixel ha associato un numero che descrive un particolare colore (o tonalità di grigio)
- Si possono definire la risoluzione (numero di punti per pollice), ed il numero di colori utilizzati

CODIFICA DI IMMAGINI

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



- Suddividiamo l'immagine mediante una griglia formata da righe orizzontali e verticali a distanza costante



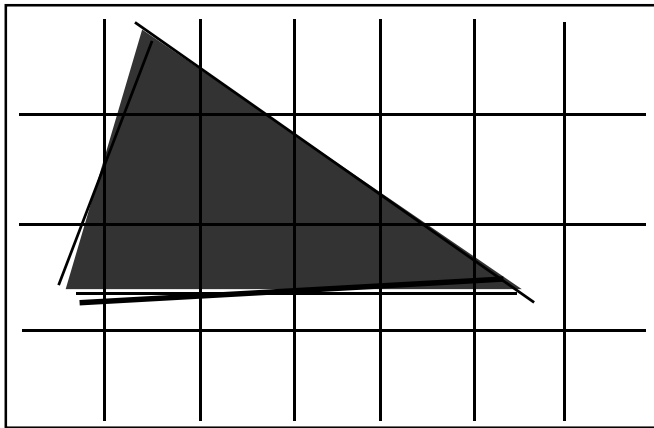
CODIFICA DI IMMAGINI

- Ogni quadratino derivante da tale suddivisione prende il nome di **pixel** (picture element) e può essere codificato in binario secondo la seguente convenzione:
 - 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

Assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



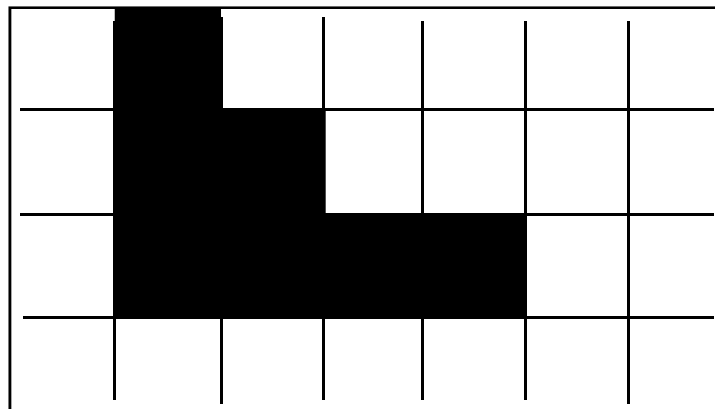
0 ₂₂	1 ₂₃	0 ₂₄	0 ₂₅	0 ₂₆	0 ₂₇	0 ₂₈
0 ₁₅	1 ₁₆	1 ₁₇	0 ₁₈	0 ₁₉	0 ₂₀	0 ₂₁
0 ₈	1 ₉	1 ₁₀	1 ₁₁	1 ₁₂	0 ₁₃	0 ₁₄
0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇

La rappresentazione della figura è data dalla stringa binaria

000000 0111100 0110000 0100000

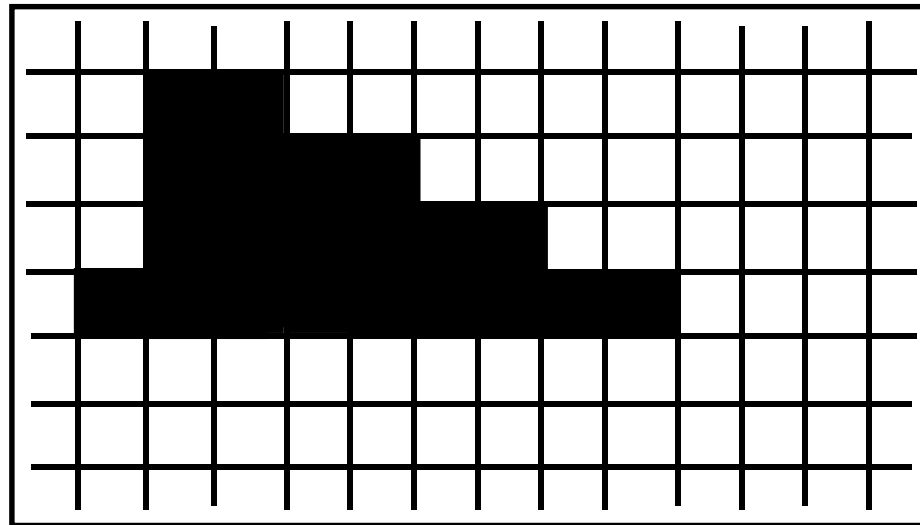
CODIFICA DI IMMAGINI

- Non sempre il contorno della figura coincide con le linee della griglia: nella codifica si ottiene un'approssimazione della figura originaria
- Se riconvertiamo la stringa **0000000011110001100000100000** in immagine otteniamo



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



CODIFICA DI IMMAGINI CON TONI DI GRIGIO

- Le immagini in bianco e nero hanno delle sfumature, o **livelli di intensità di grigio**
- Per codificare immagini con sfumature:
 - si fissa un insieme di livelli (*toni*) di grigio, cui si assegna convenzionalmente una rappresentazione binaria
 - per ogni pixel si stabilisce il livello medio di grigio e si memorizza la codifica corrispondente a tale livello
- Per memorizzare un pixel non è più sufficiente 1 bit.
 - con **4** bit si possono rappresentare **$2^4=16$** livelli di grigio
 - con **8** bit ne possiamo distinguere **$2^8=256$** ,
 - con **K** bit ne possiamo distinguere **2^K**

CODIFICA DI IMMAGINI A COLORI

- Analogamente possono essere codificate le immagini a colori:
 - bisogna definire un insieme di sfumature di colore differenti (es. RGB = Red Green Blue), codificate mediante una opportuna sequenza di bit
- La rappresentazione di un'immagine mediante la codifica dei pixel viene chiamata **codifica bitmap**

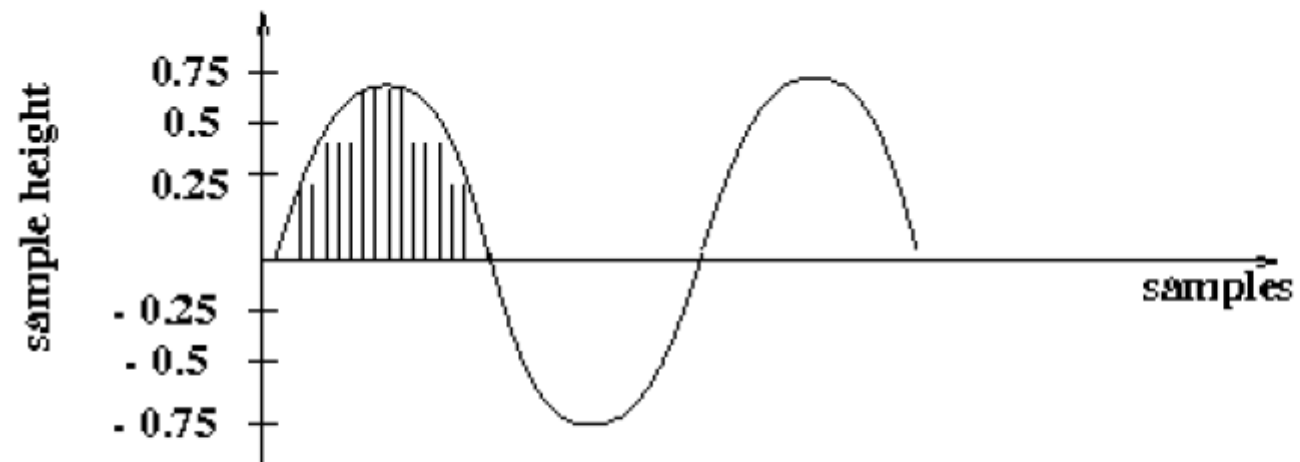
CODIFICA DI IMMAGINI A COLORI

- Il numero di byte richiesti dipende dalla **risoluzione** e dal **numero di colori** che ogni pixel può assumere
- *Es:* per distinguere **256** colori sono necessari 8 bit per la codifica di ciascun pixel
 - la codifica di un'immagine formata da 640×480 pixel richiederà 2457600 bit (307200 byte)
- 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** consentono di ridurre notevolmente lo spazio occupato dalle immagini

CODIFICA DI FILMATI

- Immagini in movimento sono memorizzate come sequenze di fotogrammi
- In genere si tratta di sequenze compresse di immagini
 - ad esempio si possono registrare solo le variazioni tra un fotogramma e l'altro
- Esistono vari formati (comprendente il sonoro):
 - *mpeg* (il più usato)
 - *avi* (microsoft)
 - *quicktime* (apple)
 - *mov*
- E' possibile ritoccare i singoli fotogrammi

CODIFICA DI SUONI



0	0.25	0.25	0.5	0.5	0.5	0.75	0.75	0.75	0.75
---	------	------	-----	-----	-----	------	------	------	------	-------