

# **Informatica per il Turismo**

## **Lezione 2: Codifica dell'Informazione**



## SOMMARIO

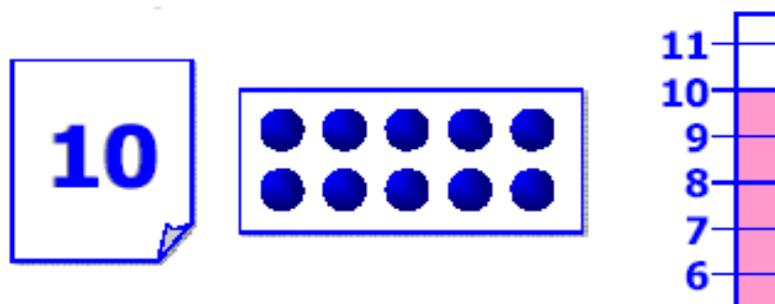
- ◆ Codifica (binaria) di informazione numerica
  - Codifica dei numeri naturali
  - Codifica dei numeri interi e razionali
- ◆ Codifica dei caratteri
- ◆ Codifica di dati multimediali
  - Compressione dell'informazione

# CODIFICA DI DATI E ISTRUZIONI

- ◆ Algoritmi
  - istruzioni che operano su dati
- ◆ Per scrivere un programma è necessario rappresentare dati e istruzioni in un formato tale che l'esecutore automatico sia in grado di
  - memorizzare istruzioni e dati
  - manipolare istruzioni e dati
- ◆ Le informazioni gestite dai sistemi di elaborazione devono essere codificate
  - per poter essere memorizzate, elaborate, scambiate,...

## CODIFICA DELL'INFORMAZIONE

- ◆ La stessa informazione si può rappresentare in modi differenti



- ◆ Stessa rappresentazione per informazioni differenti



## SISTEMI DI CODIFICA

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

## IL SISTEMA DECIMALE

### ◆ Alfabeto

- Cifre “0”, “1”, “2”, ..., “9”
- separatore decimale (“,”)
- separatore delle migliaia (“.”)
- Segni positivo (“+”) e negativo (“-”)

### ◆ Regole di composizione (**sintassi**)

- Definiscono le combinazioni ben formate
  - 12.318,43
  - 12,318.43

### ◆ Codice (**semantica**)

- Associano ad ogni configurazione un’entità di informazione
  - $12.318,43 = 1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 1 \times 10 + 1 \times 10^0 + 4 \times 10^{-1} + 3 \times 10^{-2}$

### ◆ Lo stesso alfabeto può essere usato per codici diversi

- $123,456 = 1 \times 10^2 + 2 \times 10 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3}$  [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]

## IL SISTEMA BINARIO

- ◆ Usa un alfabeto di **2** simboli
- ◆ Utilizzato nei sistemi informatici
  - Si utilizza una grandezza fisica (luminosità, tensione elettrica, la corrente elettrica), per rappresentare l'informazione
  - Si divide in due intervalli l'insieme dei valori che la grandezza può assumere: ogni intervallo corrisponde ad un simbolo
- ◆ Solo 2 simboli al fine di ridurre la probabilità di errore
  - Tanti più simboli si devono distinguere, tanto meno la rilevazione sarà affidabile (gli intervalli della grandezza fisica saranno meno ampi)

## IL SISTEMA BINARIO

### ◆ **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

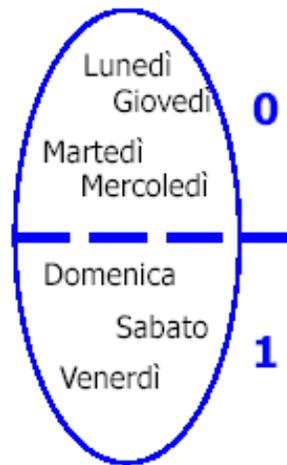
### ◆ Combinando più bit si può codificare un numero maggiore di stati

- con 2 bit possono rappresentare 4 stati
- con **K** bit si possono rappresentare  **$2^K$**  stati

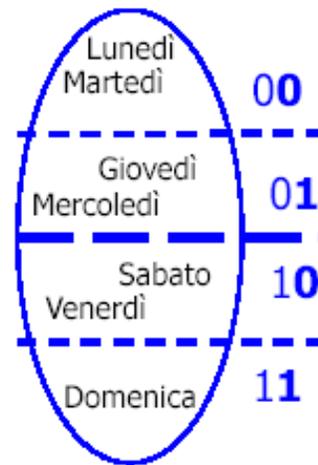
### ◆ Quanti bit sono necessari per codificare N oggetti?

- $N \leq 2^K \rightarrow K \geq \log_2 N \rightarrow \mathbf{K = \lceil \log_2 N \rceil}$

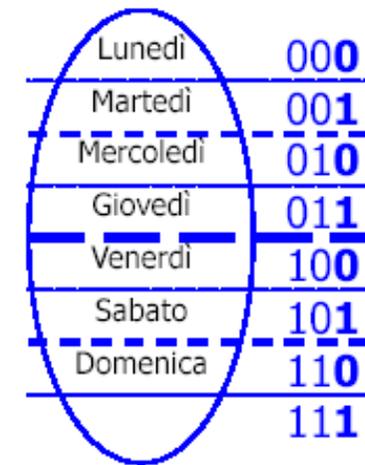
## I GIORNI DELLA SETTIMANA IN BINARIO



**1 bit**  
**2 "gruppi"**



**2 bit**  
**4 "gruppi"**



**3 bit**  
**8 "gruppi"**

## CODIFICA BINARIA: UNITA' DERIVATE

- ◆ **Byte** = 8 bit
  - può rappresentare  $2^8 = 256$  stati
- ◆ **KiloByte** (KB) =  $2^{10}$  byte = 1.024 byte  $\cong 10^3$  byte
- ◆ **MegaByte** (MB) =  $2^{20}$  byte = 1.048.576 byte  $\cong 10^6$  byte
- ◆ **GigaByte** (GB) =  $2^{30}$  byte = 1.073.741.824 byte  $\cong 10^9$  byte
- ◆ **TeraByte** (TB) =  $2^{40}$  byte = 1.099.511.627.776 byte  $\cong 10^{12}$  byte

## CODIFICA DEI NUMERI NATURALI

- ◆ Sistema di numerazione posizionale con **base  $\beta$** 
  - $\beta$  simboli (**cifre**) corrispondono ai numeri da 0 a  $\beta-1$
  - i numeri naturali possono essere rappresentati da una sequenza di cifre
- ◆ Se un numero naturale  $N$  è rappresentato in base  $\beta$  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 espresso in funzione delle potenze di 2 è pari a

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

cioè può essere rappresentato dalla seguente sequenza di bit

1 1 0 1

## CODIFICA DEI NUMERI NATURALI

### ◆ Quindi

- Numero = sequenza di bit (codifica in base 2)
- Con **K** bit si rappresentano i numeri da 0 a  **$2^K-1$**

### ◆ Esempi:

- **2** = sequenza **1 0**
- **3** = sequenza **1 1**
- **4** = sequenza **1 0 0**
- .....

## CONVERSIONE DECIMALE-BINARIO

- ◆ Si calcolano i resti della divisione per 2

$$\begin{array}{r} 18 : 2 = 9 \quad \text{resto } 0 \\ 9 : 2 = 4 \quad \text{resto } 1 \\ 4 : 2 = 2 \quad \text{resto } 0 \\ 2 : 2 = 1 \quad \text{resto } 0 \\ 1 : 2 = 0 \quad \text{resto } 1 \end{array}$$

**10010**

$$\begin{array}{r} 137 : 2 = 68 \quad \text{resto } 1 \\ 68 : 2 = 34 \quad \text{resto } 0 \\ 34 : 2 = 17 \quad \text{resto } 0 \\ 17 : 2 = 8 \quad \text{resto } 1 \\ 8 : 2 = 4 \quad \text{resto } 0 \\ 4 : 2 = 2 \quad \text{resto } 0 \\ 2 : 2 = 1 \quad \text{resto } 0 \\ 1 : 2 = 0 \quad \text{resto } 1 \end{array}$$

**10001001**

## CODIFICA DEI NUMERI INTERI

### ◆ Modulo e segno

- Il bit più a sinistra rappresenta il segno del numero  
(0 = '+', 1 = '-')
- Esempio: +7 = 0111, -7 = 1111
- Valori da  $-2^{k-1}+1$  a  $2^{k-1}-1$
- Con k=4 bit: da  $-2^3+1=-7$  a  $2^3-1=+7$
- Attenzione: ci sono due zeri!  
+0=0000 e -0=1000

## CODIFICA DEI NUMERI INTERI

Codice	Nat	MS
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7

Codice	Nat	MS
1000	8	-0
1001	9	-1
1010	10	-2
1011	11	-3
1100	12	-4
1101	13	-5
1110	14	-6
1111	15	-7

## CODIFICA DEI NUMERI INTERI

Codice	Nat	MS	C2
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7

Codice	Nat	MS	C2
1000	8	-0	-8
1001	9	-1	-7
1010	10	-2	-6
1011	11	-3	-5
1100	12	-4	-4
1101	13	-5	-3
1110	14	-6	-2
1111	15	-7	-1

## CODIFICA DEI NUMERI RAZIONALI

- Rappresentazione in **virgola fissa**
  - $0.1011_{\text{due}} = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$   
 $= 0.5 + 0.125 + 0.0625 = 0.6875_{\text{dieci}}$
  - $11.101_{\text{due}} = 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$   
 $= 2 + 1 + 0.5 + 0.125 = 3.625_{\text{dieci}}$
  - Il numero di cifre prima e dopo la “virgola” è **fisso!!**
- Rappresentazione in **virgola mobile** (float)
  - usata spesso anche in decimale per rappresentare numeri molto grandi o molto piccoli:  $0.1357 \times 10^{64}$ 
    - **mantissa** – parte frazionaria compresa tra 0 e 1 [0.1357]
    - **esponente** – numero **intero** ( $\geq 0$ )
  - utilizza 1 bit per il segno (s), h bit per l’esponente (e) e k bit per la mantissa (m):  **$R = s \times m \times 2^e$**

## CODIFICA DI CARATTERI

- ◆ Associando un simbolo dell'alfabeto ad ogni numero possiamo codificare tutte le lettere
- ◆ Codifica **ASCII** (*American Standard Code for Information Interchange*):
  - Caratteri speciali, punteggiatura, a-z, A-Z, 0-9
  - Utilizza 7 bit (128 caratteri)
  - I codici ASCII estesi usano 8 bit (256 caratteri)
- ◆ Codifica **UNICODE**
  - Utilizza 16 bit (65536 caratteri)
  - I primi 128 caratteri di UNICODE sono gli stessi di ASCII
  - I successivi corrispondono ad altri alfabeti (greco, cirillico, ebraico, ...)
  - Non riesce a coprire i simboli (oltre 200000) di tutte le lingue!

## ASCII SU 7 BIT

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
010	sp	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
110	`	a	b	c	d	e	f	g	h	I	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	Y	z	{		}	~	canc

“Ciao” = 1000011 1101001 1100001 1101111

“24” = 0110001 0110011

“3 kg” = 0110011 0100000 1101011 1100111

## CODIFICA DI DATI MULTIMEDIALI

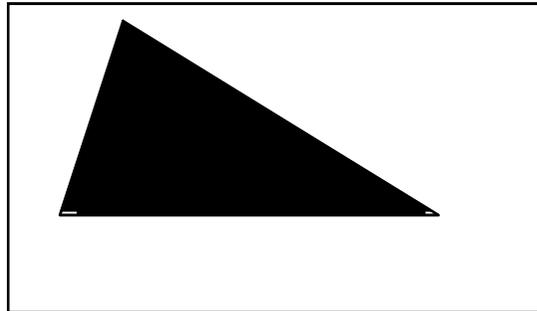
- ◆ Lettere e numeri non costituiscono le uniche informazioni utilizzate dagli elaboratori. Le applicazioni usano ed elaborano anche altri tipi di informazione:
  - diagrammi
  - immagini
  - suoni
- ◆ Spesso in questi casi si parla di applicazioni di tipo **multimediale**

## CODIFICA DI IMMAGINI

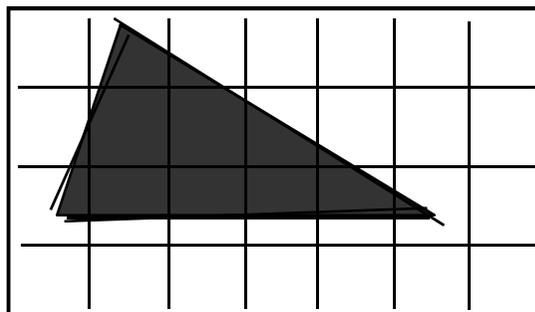
- ◆ Esistono numerose tecniche per la memorizzazione digitale e l'elaborazione di un'immagine
- ◆ Immagini = sequenze di bit!
- ◆ L'immagine viene **digitalizzata** cioè rappresentata con sequenze di pixel
- ◆ Ogni pixel ha associato un numero che descrive un particolare colore (o tonalità di grigio)

## 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

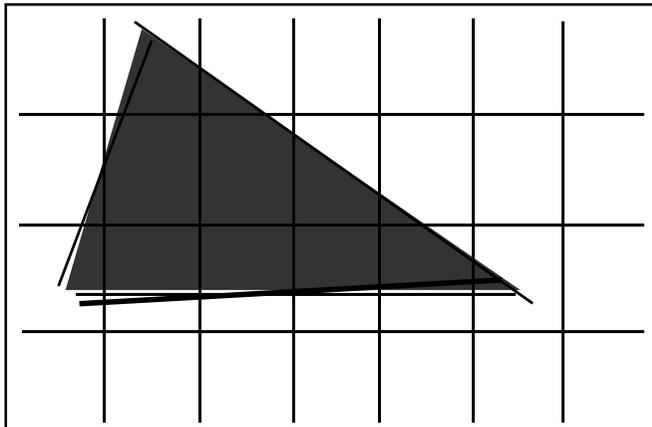


## CODIFICA DI IMMAGINI

- ◆ 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
- ◆ **ipotesi:** assumiamo che i pixel siano ordinati dal basso verso l'alto e da sinistra verso destra



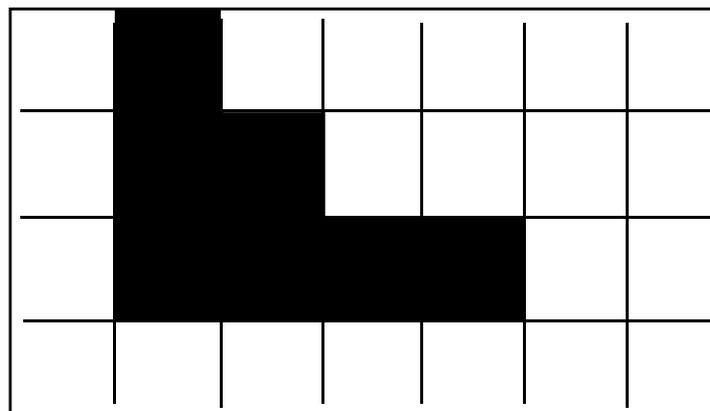
0 <sub>22</sub>	1 <sub>23</sub>	0 <sub>24</sub>	0 <sub>25</sub>	0 <sub>26</sub>	0 <sub>27</sub>	0 <sub>28</sub>
0 <sub>15</sub>	1 <sub>16</sub>	1 <sub>17</sub>	0 <sub>18</sub>	0 <sub>19</sub>	0 <sub>20</sub>	0 <sub>21</sub>
0 <sub>8</sub>	1 <sub>9</sub>	1 <sub>10</sub>	1 <sub>11</sub>	1 <sub>12</sub>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

La rappresentazione della figura è data dalla stringa binaria

**000000 0111100 0110000 0100000**

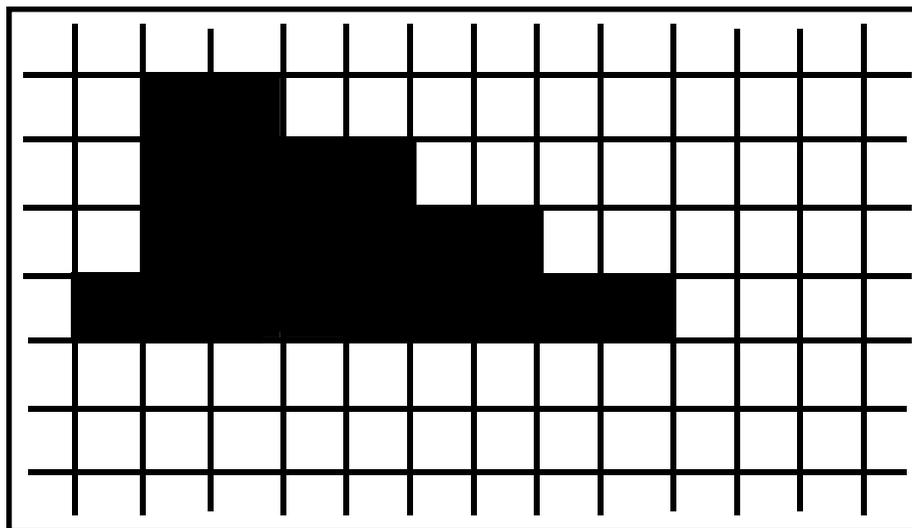
## CODIFICA DI IMMAGINI

- ◆ 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, 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  $640 \times 480$  pixel richiederà 2457600 bit (307200 byte)
- ◆ I monitor tipici utilizzano
  - risoluzione:  $640 \times 480$ ,  $1024 \times 768$ ,  $1280 \times 1024$
  - 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 piu' usato)
  - *avi* (microsoft)
  - *quicktime* (apple)
  - *mov, ...*

## CODIFICA DI SUONI

- ◆ L'onda sonora viene misurata (**campionata**) ad intervalli regolari
- ◆ Minore è l'intervallo di campionamento e maggiore è la qualità del suono
- ◆ CD musicali: 44000 campionamenti al secondo, 16 bit per campione.
- ◆ Alcuni formati:
  - *.mov, .wav, .mpeg, .avi, .midi*

## COMPRESSIONE DEI DATI

- ◆ Vantaggi: memorizzazione e trasmissione
- ◆ Lossless
  - Senza perdita di informazione
  - Necessario per informazione di tipo testuale (programmi, documenti)
- ◆ Lossy
  - Con perdita di informazione
  - Rapporto di compressione variabile dall'utente
  - Immagini: GIF, JPEG (elimina lievi cambiamenti di colore)
  - Animazioni: MPEG (memorizza solo differenze tra fotogrammi)
  - Audio: MP3 (elimina suoni a basso volume sovrapposti con suoni ad alto volume)

## JPEG: Fattore qualità 90/100 (253KB)

800x600

16,8mln  
colori  
24 bit

Bitmap:  
1440000  
byte

JPEG:  
258971  
byte



## JPEG: Fattore qualità 50/100 (30KB)



## JPEG: Fattore qualità 25/100 (20KB)



## JPEG: Fattore qualità 10/100 (12KB)



## JPEG: Fattore qualità 1/100 (9KB)

