

# I codici

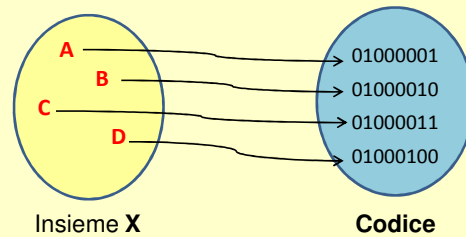
Un **codice** è costituito da:

- un **alfabeto** che è un insieme di **simboli o segni**, (es. alfabeto binario {0,1} )
- un insieme di **parole** che sono costruite a partire dai simboli dell'alfabeto (es. 01000001, 01000010, .....

Per poter utilizzare un codice deve essere definito un **processo di codifica**

che consiste

**nell'associare i singoli oggetti di un insieme X a ciascun simbolo o parola del codice**



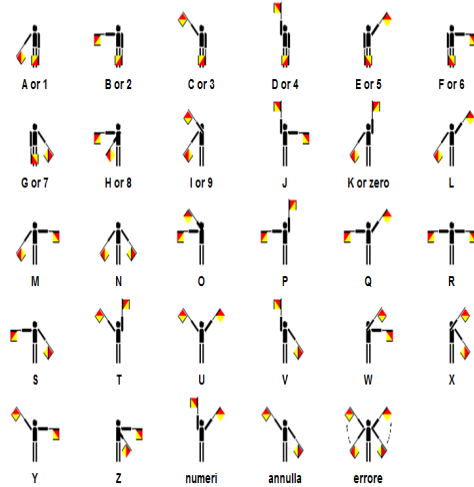
### International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	• —	U	••• —
B	•••••	V	••• —•
C	•• —•••	W	•• —••
D	••• —•	X	••••• —
E	•••••	Y	••••• —•
F	••••• —•	Z	••••• —••
G	••••• —••	1	••••• —•••
H	••••• —•••	2	••••• —••••
I	••••• —••••	3	••••• —•••••
J	••••• —•••••	4	••••• —••••••
K	••••• —••••••	5	••••• —•••••••
L	••••• —•••••••	6	••••• —••••••••
M	••••• —••••••••	7	••••• —•••••••••
N	••••• —•••••••••	8	••••• —••••••••••
O	••••• —••••••••••	9	••••• —•••••••••••
P	••••• —•••••••••••	0	••••• —••••••••••••
Q	••••• —••••••••••••		
R	••••• —•••••••••••••		
S	••••• —••••••••••••••		
T	••••• —•••••••••••••••		

### segnalazioni con bandiere a mano

Il sistema di segnalazione con bandiere è un alfabeto basato sulla segnalazione per mezzo dello sventolamento di due bandiere tenute in mano



### Tabella ASCII Standard

DECIMALE	BINARIO	ESADECIMALE	ASCII
0	00000000	0	nul
1	00000001	1	soh
2	00000010	2	stx
3	00000011	3	etx
4	00000100	4	eot
5	00000101	5	enq
6	00000110	6	ack
7	00000111	7	bel
8	00001000	8	bs
9	00001001	9	ht
10	00001010	A	ff
11	00001011	B	vi
12	00001100	C	ff
13	00001101	D	cr
14	00001110	E	so
15	00001111	F	si
16	00010000	10	dlh
17	00010001	11	dc1
18	00010010	12	dc2
19	00010011	13	dc3
20	00010100	14	dc4
21	00010101	15	nak
22	00010110	16	syn
23	00010111	17	etb
24	00011000	18	can
25	00011001	19	em
26	00011010	1A	sub
27	00011011	1B	esc
28	00011100	1C	fs
29	00011101	1D	gs
30	00011110	1E	rs
31	00011111	1F	us
32	00100000	20	spazio
33	00100001	21	!
34	00100010	22	@
35	00100011	23	#
36	00100100	24	\$
37	00100101	25	%
38	00100110	26	&
39	00100111	27	'
40	00101000	28	(
41	00101001	29	)
42	00101010	2A	{

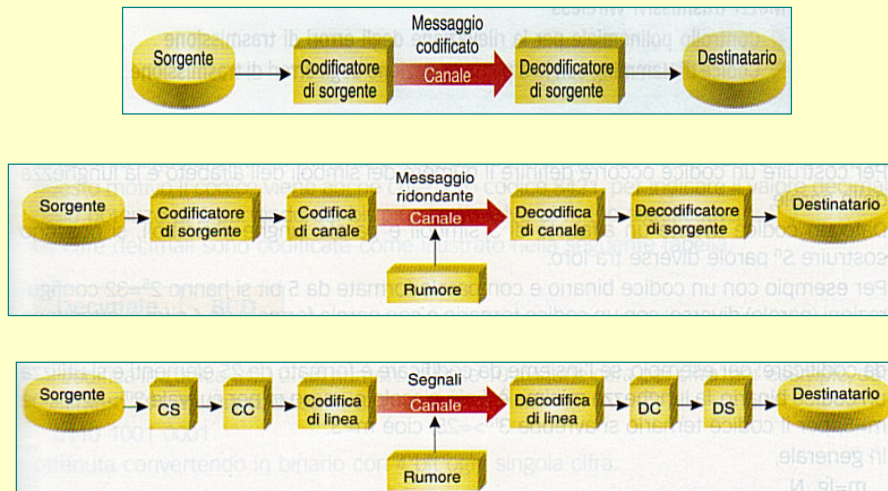
DECIMALE	BINARIO	ESADECIMALE	ASCII
43	00101011	2B	+
44	00101100	2C	,
45	00101101	2D	-
46	00101110	2E	.
47	00101111	2F	/
48	00110000	30	0
49	00110001	31	1
50	00110010	32	2
51	00110011	33	3
52	00110100	34	4
53	00110101	35	5
54	00110110	36	6
55	00110111	37	7
56	00111000	38	8
57	00111001	39	9
58	00111010	3A	:
59	00111011	3B	;
60	00111100	3C	<
61	00111101	3D	=
62	00111110	3E	>
63	00111111	3F	?
64	01000000	40	@
65	01000001	41	A
66	01000010	42	B
67	01000011	43	C
68	01000100	44	D
69	01000101	45	E
70	01000110	46	F
71	01000111	47	G
72	01001000	48	H
73	01001001	49	I
74	01001010	4A	J
75	01001011	4B	K
76	01001100	4C	L
77	01001101	4D	M
78	01001110	4E	N
79	01001111	4F	O
80	01010000	50	P
81	01010001	51	Q
82	01010010	52	R
83	01010011	53	S
84	01010100	54	T
85	01010101	55	U

DECIMALE	BINARIO	ESADECIMALE	ASCII
86	01010110	56	V
87	01010111	57	W
88	01011000	58	X
89	01011001	59	Y
90	01011010	5A	Z
91	01011011	5B	[
92	01011100	5C	\
93	01011101	5D	]
94	01011110	5E	^
95	01011111	5F	_
96	01100000	60	`
97	01100001	61	a
98	01100010	62	b
99	01100011	63	c
100	01100100	64	d
101	01100101	65	e
102	01100110	66	f
103	01100111	67	g
104	01101000	68	h
105	01101001	69	i
106	01101010	6A	j
107	01101011	6B	k
108	01101100	6C	l
109	01101101	6D	m
110	01101110	6E	n
111	01101111	6F	o
112	01110000	70	p
113	01110001	71	q
114	01110010	72	r
115	01110011	73	s
116	01110100	74	t
117	01110101	75	u
118	01110110	76	v
119	01110111	77	w
120	01111000	78	x
121	01111001	79	y
122	01111010	7A	z
123	01111011	7B	{
124	01111100	7C	
125	01111101	7D	}
126	01111110	7E	~
127	01111111	7F	`

# I codici rilevatori e correttori di errori

5

## Modello di un processo di comunicazione



6



In una trasmissione dati la stazione **mittente** aggiunge ai dati da trasmettere un **codice di controllo** per riconoscere se si sono verificati degli **errori durante la trasmissione**.

La stazione **ricevente** usa questo codice per **individuare ed eventualmente correggere** l'errore.

I **codici rilevatori** permettono soltanto di capire che i dati non sono corretti.

I **codici correttori** permettono non solo di capire che si è verificato un errore, ma anche di **individuare la posizione dell'errore** e quindi di **correggerlo**.

I **codici correttori** richiedono **molti più bit** dei **codici rilevatori** e quindi sprecano ampiezza di banda. Per questo di solito perciò si usano codici rilevatori.

7

## • Codici ridondanti

Per consentire la **rilevazione** e la **correzione di errori**

si ricorre frequentemente a **codici ridondanti**

ovvero codici che **utilizzano un numero maggiore di bit rispetto al numero strettamente necessario** per codificare le informazioni dell'insieme sorgente.

Ad esempio:

$m$  bit di dati (l'informazione da trasmettere)

$r$  bit di controllo (bit ridondanti)



ciascuna parola del codice utilizza  
 $n = m + r$  bit

**101001010110** Il codice **non è efficiente**: il 33% dei bit sono ridondanti

8

**Codici ridondanti** vengono usati per esempio:

- nei **sistemi di comunicazione**
  - nelle **operazioni di memorizzazione dei dati**.
- Nei **sistemi di comunicazione** ciò permette di verificare se le informazioni sono state trasmesse correttamente.  
Se viene usato un codice rilevatore, in caso di errore bisogna chiedere al mittente di ritrasmettere le informazioni.
- In una **operazione di memorizzazione** l'errore si può verificare al momento della **scrittura** o della successiva **lettura** dei dati.  
Se viene rilevato un errore si può ritentare la lettura, ma se l'errore si è verificato in scrittura il problema si può risolvere solo se è stato usato un codice correttore.

### La cifra di controllo nell'ISBN-13

Il relativo algoritmo numerico è il seguente:

- Partendo da sinistra, **si moltiplica ogni cifra per un peso** in base alla posizione della cifra stessa: la prima cifra per **1**, la seconda per **3** la terza per **1** e così via....
- **si sommano i risultati**
- **si divide la somma per 10** e si prende il **resto della divisione**
- **si sottrae il risultato da 10: questa cifra è la cifra di controllo**

Esempio: dato un codice **978-88-430-2534 ?**

$$9 \cdot 1 + 7 \cdot 3 + 8 \cdot 1 + 8 \cdot 3 + 8 \cdot 1 + 4 \cdot 3 + 3 \cdot 1 + 0 \cdot 3 + 2 \cdot 1 + 5 \cdot 3 + 3 \cdot 1 + 4 \cdot 3 = 117$$

$$\frac{117}{10} = 11 \quad \text{col resto di } 7$$

$$10 - 7 = 3 \quad \text{quindi } \mathbf{3} \text{ è la 13-esima cifra da aggiungere}$$

**ISBN 978-88-430-2534 3**

La **prova della correttezza** di un codice ISBN-13 si ottiene **rieseguendo il calcolo iniziale**, stavolta con tutte e 13 le cifre. Il risultato dell'addizione dovrà essere divisibile per 10.

$$9 \cdot 1 + 7 \cdot 3 + 8 \cdot 1 + 8 \cdot 3 + 8 \cdot 1 + 4 \cdot 3 + 3 \cdot 1 + 0 \cdot 3 + 2 \cdot 1 + 5 \cdot 3 + 3 \cdot 1 + 4 \cdot 3 + 3 \cdot 1 = 120$$

## • Distanza di Hamming

Il numero di bit diversi tra due parole di codice viene detto **distanza di Hamming**.

Si può calcolare il numero di bit diversi facendo l'**or esclusivo** delle due stringhe e contando il numero di bit 1 del risultato.

Se due parole hanno distanza  $d$  significa che servono  $d$  **errori** per trasformare una nell'altra.

es.: le stringhe: **1011011**    **1010010** hanno **distanza 2**  
perché hanno solo 2 bit diversi

## Calcolo della distanza di Hamming di un codice

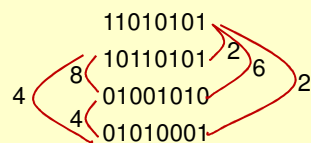
Un codice è costituito da un insieme di parole "legali".

- Per ogni coppia di parole del codice si calcola la distanza di Hamming
- La **distanza minima**  $D$  tra le parole legali del codice è la **distanza di Hamming del codice**.

Esempio: codice con  $N=4$  parole legali.

Il numero delle possibili coppie è dato dalle **combinazioni semplici** di  $N$  elementi presi a  $K$  a  $K$  (nel nostro caso  $K=2$ )

$$\binom{N}{K} = \frac{N!}{K!(N-K)!}$$



La **distanza di Hamming del codice** è  $D=2$

La distanza di Hamming indica quanti errori si possono **rilevare** e quanti se ne possono **correggere**.

**Teorema 1:** condizione necessaria e sufficiente affinché un codice sia in grado di **rilevare  $k$  errori** è che la distanza di Hamming del codice sia:

$$D = k + 1;$$

**Teorema 2:** condizione necessaria e sufficiente affinché un codice sia in grado di **correggere  $k$  errori** è che la distanza di Hamming del codice sia:

$$D = 2k + 1;$$

### Codici rilevatori

- **Bit di parità semplice** (pari o dispari)
- **Codice CRC** (a ridondanza ciclica)

### Codici correttori

- **Bit di parità incrociata** (trasversale e longitudinale)
- **Codice di Hamming**