



# La rappresentazione delle Informazioni

---

- Nella vita di tutti i giorni siamo abituati ad avere a che fare con vari tipi di informazioni, di natura e forma diversa, così come siamo abituati a diverse rappresentazioni della stessa informazione

Esempio: un testo può essere in forma stampata oppure la stessa informazione può essere registrata su un nastro

- Rappresentazioni *equivalenti* della stessa informazione e meccanismi di trasformazione tra differenti rappresentazioni
- La scelta della rappresentazione è in genere vincolata al tipo di utilizzo ed al tipo di operazioni che devono essere fatte sulle informazioni stesse



# La rappresentazione delle informazioni

---

- I computer memorizzano ed elaborano le informazioni sotto forma di **bit (Binary Digit)**
- Un bit è l'unità di informazione base e può rappresentare **due** informazioni:
  - vero o falso
  - acceso o spento
  - .....
- Rappresentazione **binaria** (o **digitale**). Il linguaggio di base mediante il quale ogni informazione deve essere codificata è costituito da due soli simboli (**0** e **1**)



# La rappresentazione delle informazioni

---

- Rappresentazione **binaria** (o **digitale**). Il linguaggio di base mediante il quale ogni informazione deve essere codificata è costituito da due soli **simboli** (**0** e **1**)
- NOTA BENE: i *simboli* 0 e 1 NON devono essere confusi con i *numeri* 0 e 1 (la tecnica per rappresentare i numeri in formato digitale verrà studiata più avanti), con le *cifre del sistema decimale* 0 e 1 (ricordo che il concetto di cifra e quello di numero sono diversi), con i *caratteri* 0 e 1 (i caratteri sono usati per la rappresentazione di testi, come vedremo più avanti)



# La rappresentazione delle informazioni

---

Le ragioni di questa scelta sono prevalentemente di tipo tecnologico e i due simboli corrispondono a:

- due stati di polarizzazione di una sostanza magnetizzabile;
- due stati di carica elettrica di una sostanza;
- al passaggio/non passaggio di corrente attraverso un cavo conduttore;
- al passaggio/non passaggio di luce attraverso un cavo ottico



## Codifica binaria

---

- Per poter rappresentare un numero maggiore di informazioni è necessario utilizzare sequenze di bit.

Per esempio, per rappresentare quattro informazioni diverse possiamo utilizzare due bit che ci permettono di ottenere quattro configurazioni distinte

00    01    10    11

- Il processo secondo cui si fa corrispondere ad un'informazione una configurazione di bit prende il nome di **codifica dell'informazione**

**Esempio:** un esame può avere quattro possibili esiti: ottimo, discreto, sufficiente, insufficiente



# Codifica binaria

- Codifico

*ottimo con*                    *00*

*discreto con*                *01*

*sufficiente con*            *10*

*insufficiente con*        *11*

- Con N bit si possono codificare  $2^N$  informazioni differenti

N	Informazioni
2	4
3	8
4	16
5	32
6	64
7	128
8	256



## Codifica binaria

- Se invece il mio problema è quello di dover rappresentare  $M$  informazioni differenti devo selezionare il numero di  $N$  bit in modo tale che

$$2^N \geq M$$

**Esempio:** per rappresentare 40 informazioni differenti devo utilizzare 6 bit perché  $2^6 = 64$ , 5 bit non sono sufficienti perché  $2^5 = 32$

- Esiste una particolare aggregazione di bit che è costituita da 8 bit ( $2^8 = 256$  informazioni) e prende il nome di **byte**



# La rappresentazione delle Informazioni

---

- Noi vedremo le tecniche per rappresentare in formato digitale:
  - i caratteri
  - i numeri naturali
  - le immagini fisse
  - il suono
  - le immagini in movimento
- Parleremo di come si procede per ottenere una rappresentazione digitale (cioè, usando bit) mentre vedremo nella parte dedicata all'architettura di un calcolatore quali sono i dispositivi che materialmente realizzano le tecniche descritte.



## La codifica dei caratteri

---

- L'obiettivo è quello di comunicare con il calcolatore usando il nostro linguaggio. Dobbiamo rappresentare le lettere dell'alfabeto
- L'insieme di simboli comunemente usati nell'alfabeto anglosassone, incluse le cifre numeriche, lettere maiuscole e minuscole, simboli di punteggiatura, parentesi e operatori aritmetici, può essere codificato usando 7 bit ( $2^7 = 128$ )
- Il metodo di codifica più diffuso tra i produttori di hardware e di software prende il nome di codice **ASCII** (**American Standard Code for Information Interchange**)



# Il codice ASCII

---

ASCII	Simb.	ASCII	Simb.	ASCII	Simb.
0000000	NUL	0001110	SO	0011100	FS
0000001	SOH	0001111	SI	0011101	GS
0000010	STX	0010000	DLE	0011110	RS
0000011	ETX	0010001	DC1	0011111	US
0000100	EOT	0010010	DC2	0100000	SP
0000101	ENQ	0010011	DC3	0100001	!
0000110	ACK	0010011	DC4	0100010	"
0000111	BEL	0010101	NAK	0100011	#
0001000	BS	0010110	SYN	0100100	\$
0001001	HT	0010111	ETB	0100101	%
0001010	NL	0011000	CAN	0100110	&
0001011	VT	0011001	EM	0100111	'
0001100	NP	0011010	SUB	0101000	(
0001101	CR	0011011	ESC	0101001	)



# Il codice ASCII

---

ASCII	Simb.	ASCII	Simb.	ASCII	Simb.
0101010	*	0111001	9	1000111	G
0101011	+	0111010	:	1001000	H
0101100	,	0111011	;	1001001	I
0101101	-	0111100	<	1001010	J
0101110	.	0111101	=	1001011	K
0101111	/	0111110	>	1001100	L
0110000	0	0111111	?	1001101	M
0110001	1	1000000	@	1001110	N
0110010	2	1000001	A	1001111	O
0110011	3	1000010	B	1010000	P
0110100	4	1000011	C	1010001	Q
0110101	5	1000100	D	1010010	R
0110110	6	1000101	E	1010011	S
0111000	8	1000110	F	1010100	T



# Il codice ASCII

---

ASCII	Simb.	ASCII	Simb.	ASCII	Simb.
1010101	U	1100011	c	1110001	q
1010110	V	1100100	d	1110010	r
1010111	W	1100101	e	1110011	s
1011000	X	1100110	f	1110100	t
1011001	Y	1100111	g	1110101	u
1011010	Z	1101000	h	1110110	v
1011011	[	1101001	i	1110111	w
1011100	\	1101010	j	1111000	x
1011101	]	1101011	k	1111001	y
1011110	^	1101100	l	1111010	z
1011111	_	1101101	m	1111011	{
1100000	`	1101110	n	1111100	
1100001	a	1101111	o	1111101	}
1100010	b	1110000	p	1111110	~
1111111	DEL				



## Il codice ASCII

---

Sebbene 7 bit siano sufficienti per codificare l'insieme di caratteri di uso comune, in genere il codice ASCII standard utilizza 8 bit, il primo dei quali è sempre 0

Codifica della parola **cane**

<b>01100011</b>	<b>01100001</b>	<b>01101110</b>	<b>01100101</b>
<b>c</b>	<b>a</b>	<b>n</b>	<b>e</b>

*Tra i simboli speciali del codice ASCII vi è anche il simbolo spazio bianco "NUL"(codice 00000000), il simbolo di fine riga "CR" (00001101)*

*In questo modo è possibile rappresentare mediante una sequenza di codici ASCII un testo strutturato in righe e pagine*



## Il codice ASCII

---

Consideriamo il problema inverso: data una sequenza di bit, il testo che essa codifica può essere ottenuto nel modo seguente:

- si divide la sequenza in gruppi di otto bit (un byte);
- si determina il carattere corrispondente ad ogni byte

Esempio:

01101001 01101100 00000000 01110000 01101111 00101110

01101001 01101100 00000000 01110000 01101111 00101110

i

l

P

o

.



## La codifica dei caratteri

---

- 52 lettere alfabetiche maiuscole e minuscole
- 10 cifre (0, 1, 2, ..., 9)
- Segni di punteggiatura (, . ; : ! " ? ' ^ \ ...)
- Segni matematici (+, -, ×, ±, {, [, >, ...)
- Caratteri nazionali (à, è, ì, ò, ù, ç, ñ, ö, ...)
- Altri segni grafici (©, ←, ↑, ‡, @, €, ...)
- In totale 220 caratteri circa



## La codifica dei caratteri

---

- Codifiche standard:
  - **ASCII**, 8 bit per carattere, rappresenta 256 caratteri
  - **UNICODE**, 16 bit per carattere, (ASCII + caratteri etnici)
- Codifiche proprietarie:
  - **MSWindows**, 16 bit per carattere, simile ad UNICODE



## La codifica dei numeri

---

- La rappresentazione dei numeri con il sistema decimale può essere utilizzata come spunto per definire un metodo di codifica dei numeri all'interno degli elaboratori: la sequenza 15 viene interpretato come: 1 decina + 5 unità
- In generale la sequenza  $c_n c_{n-1} c_{n-2} \dots c_1 c_0$  (ogni " $c_i$ " è una cifra compresa tra "0" e "9") viene interpretata come:

$$\begin{aligned} &c_0 \times 10^0 + && (c_0 \text{ unità}) \\ &c_1 \times 10^1 + && (c_1 \text{ decine}) \\ &c_2 \times 10^2 + && (c_2 \text{ centinaia}) \\ & && \dots \\ &c_{n-1} \times 10^{n-1} + \\ &c_n \times 10^n \end{aligned}$$



## Un ripasso di aritmetica: la notazione posizionale

---

- La numerazione decimale utilizza una **notazione** posizionale basata sul numero 10 (**base**). La sequenza "234" rappresenta il numero  $4 \times 10^0 + 3 \times 10^1 + 2 \times 10^2$
- La notazione posizionale può essere utilizzata in qualunque altro sistema di numerazione (con base diversa da 10)
- Nel sistema di numerazione binario i numeri vengono codificati utilizzando le due cifre "0" e "1"
- Nel sistema di numerazione ottale i numeri vengono codificati utilizzando le otto cifre "0", "1", ....., "7"
- Nel sistema di numerazione esadecimale i numeri vengono codificati utilizzando le sedici cifre "0", "1", ....., "8", "9", "A", "B", "C", "D", "E", "F"



## Un ripasso di aritmetica: La notazione posizionale

- In analogia con il caso decimale la sequenza  $c_n c_{n-1} c_{n-2} \dots c_1 c_0$  (ogni " $c_i$ " è la cifra "0" o la cifra "1") rappresenterà, in binario, il numero

$$c_0 \times 2^0 + c_1 \times 2^1 + \dots c_{n-1} \times 2^{n-1} + c_n \times 2^n$$

La sequenza "1011"denota il numero

$$1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 11 \text{ (in base 10)}$$

- In analogia con il caso decimale la sequenza  $c_n c_{n-1} c_{n-2} \dots c_1 c_0$  rappresenterà, in esadecimale, il numero

$$c_0 \times 16^0 + c_1 \times 16^1 + \dots c_{n-1} \times 16^{n-1} + c_n \times 16^n$$

La sequenza "1011"denota il numero

$$1 \times 16^0 + 1 \times 16^1 + 0 \times 16^2 + 1 \times 16^3 = 4113 \text{ (in base 10)}$$

*Per evitare ambiguità si usa la notazione  $1011_2 = 11_{10}$*

*Per evitare ambiguità si usa la notazione  $1011_{16} = 4113_{10}$*



## Un ripasso di aritmetica: Rappresentazione decimale - limitazioni

- Consideriamo la base dieci: con tre cifre decimali si possono rappresentare i numeri compresi tra 0 e 999, il numero successivo (1000) richiede una quarta cifra di cui non disponiamo

In questo caso si dice che si ha un problema di **overflow**, ossia si eccede il numero di cifre destinato alla rappresentazione, e si genera un errore perché il numero non può essere gestito

Poiché il numero 999 può essere scritto come  $10^3-1$  (ossia 1000-1), possiamo enunciare la seguente regola:

con N cifre decimali si possono rappresentare  
i numeri da 0 a  $10^N-1$



## Un ripasso di aritmetica: Rappresentazione binaria - limitazioni

Consideriamo la base due: con tre cifre binarie si possono rappresentare i numeri compresi tra 0 e  $2^3-1$  (ossia 8-1), possiamo enunciare la seguente regola:

**con N cifre binarie si possono rappresentare i numeri da  
0 a  $2^N-1$**

Esempio con  $N=3$ :

numero decimale	rappresentazione binaria
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



## Un ripasso di aritmetica: Rappresentazione esadecimale - limitazioni

Consideriamo la base sedici: con tre cifre esadecimali si possono rappresentare i numeri compresi tra 0 e  $16^3-1$  (ossia 4096-1).

**con N cifre esadecimali si possono rappresentare i numeri da 0 a  $16^N-1$**

Esempio con N=2:

numero decimale	rappresentazione esadecimale
0	00
1	01
.....	.....
10	0A
11	0B
.....	.....
15	0F
16	10
17	11
.....	.....
30	1E
31	1F
32	20
.....	.....



## Un ripasso di aritmetica: Rappresentazione binaria - operazioni

---

- A queste rappresentazioni si possono applicare le operazioni aritmetiche:

$$0+0=0$$

$$1+0=1$$

$$0+1=1$$

$1+1=0$  con riporto di 1 ovvero 10

- $1+1$  in decimale è uguale a 2 ma siamo nella notazione binaria che ha solo due cifre, 0 e 1