La digitalizzazione delle informazioni

(come e perché trasformare le informazioni in numeri)

A cura di Saverio Cantone

Digitalizzare

•I computer possono riconoscere e gestire **solo** la corrente elettrica e non capiscono nulla di numeri, lettere, colori, suoni e filmati.

•Per consentire alle macchine di “gestire” tali informazioni occorre prima ***digitalizzare*** le informazioni.

•Digitalizzare significa tradurre in **DIGIT** (digit=cifra) ossia trasformare in cifre.

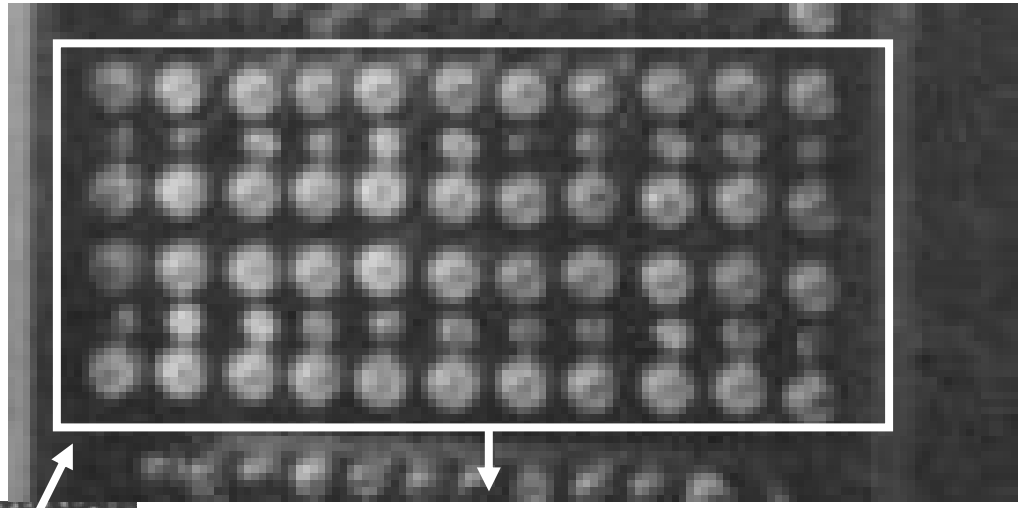
•Tali cifre possono essere solo 0 e 1, lo zero sarà associato all’assenza di corrente e 1 alla presenza.

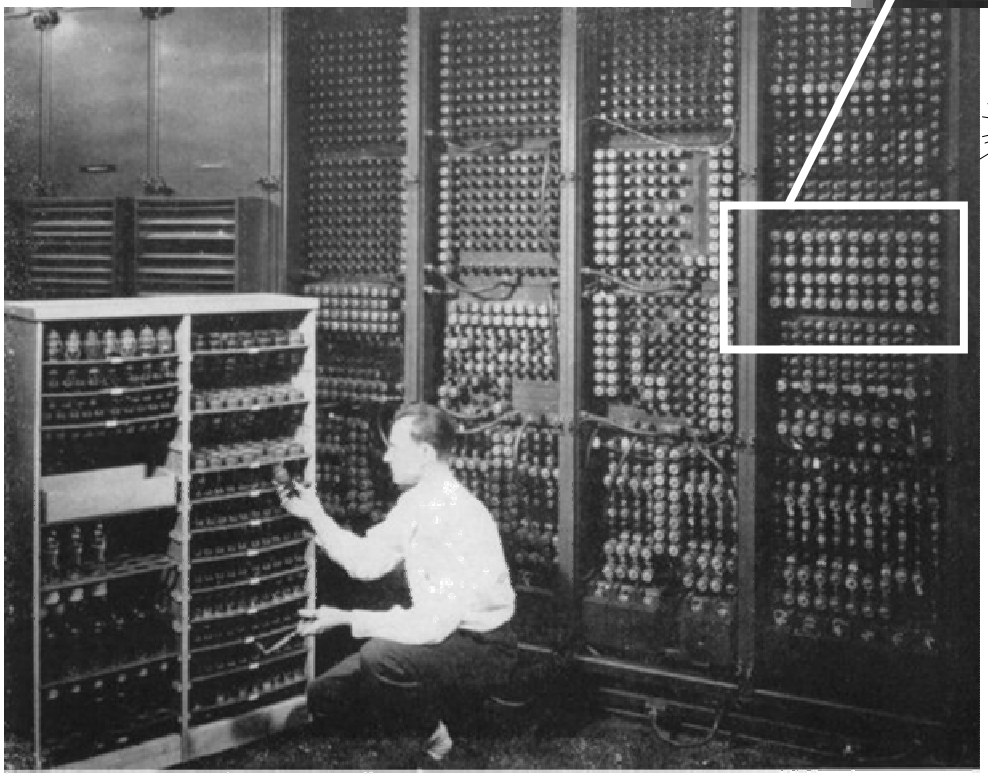
 ⟺ **0**

 ⟺ **1**

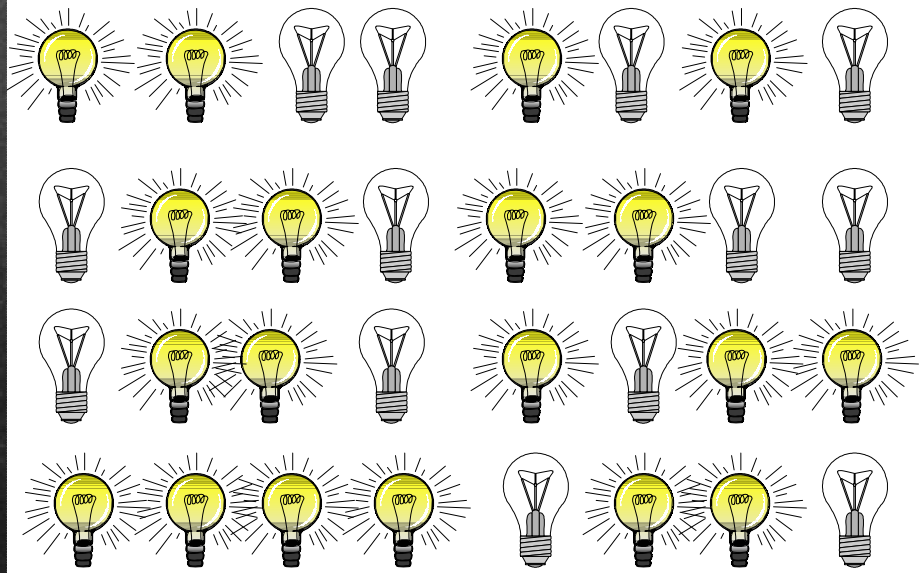
Le valvole

Nell'ENIAC (1944) ben 19.000 valvole consentivano di gestire 19.000 informazioni digitali



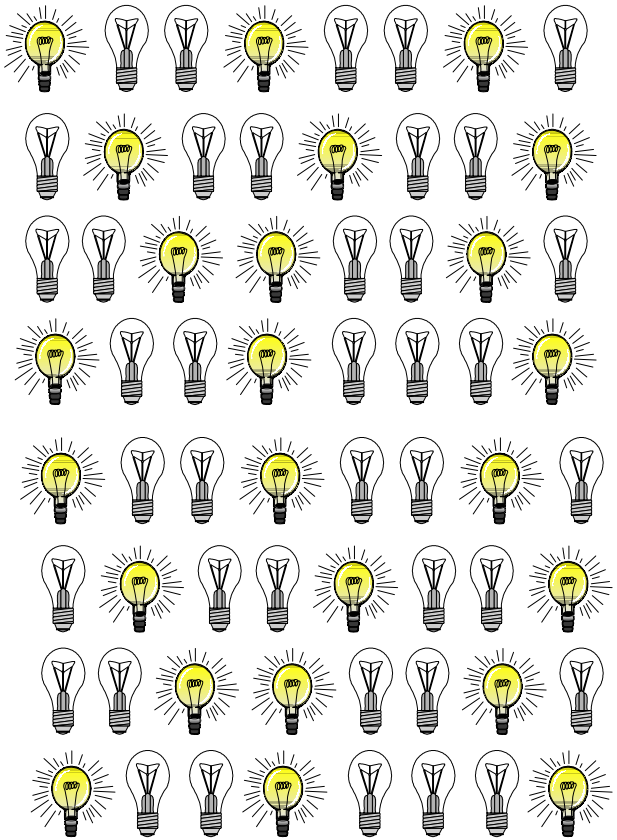


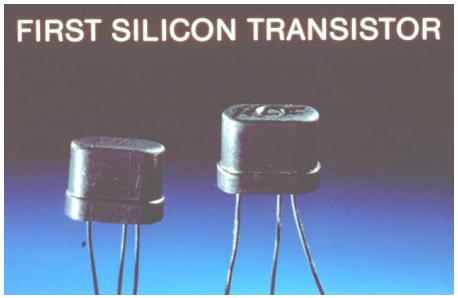
Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.



I transistor

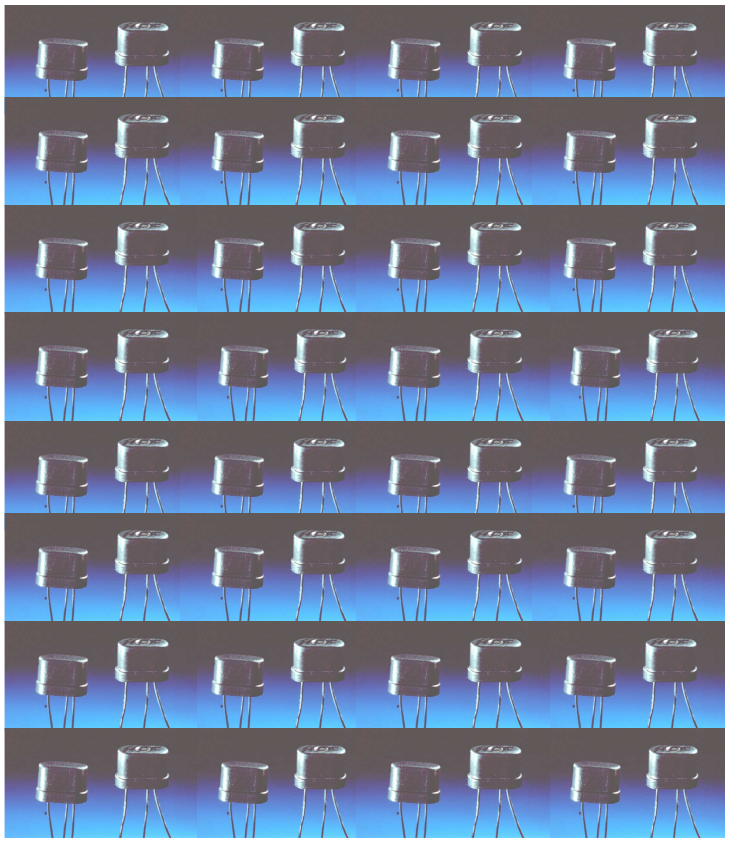
Grazie ai progressi dell'elettronica, la funzione di molte valvole viene svolta da pochi transistor…

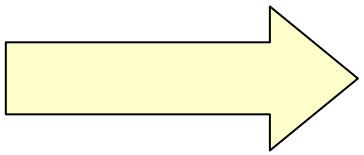




I circuiti integrati

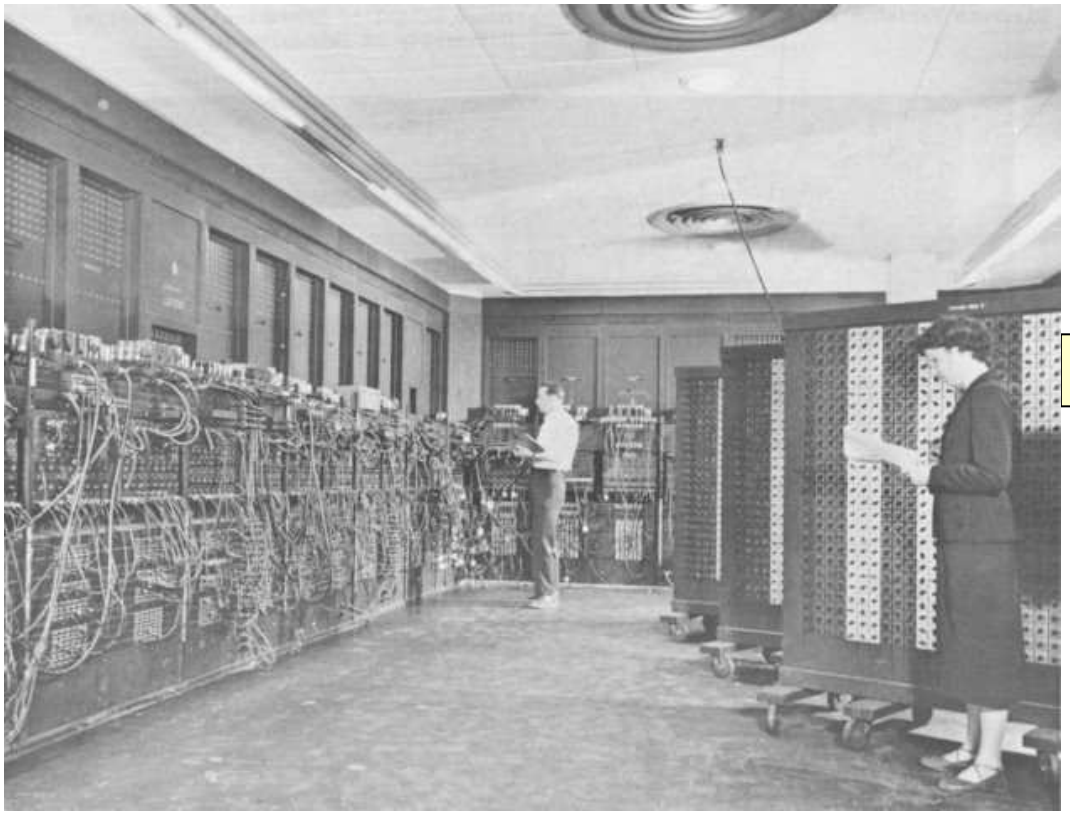
…e la funzione di milioni di transistor viene svolta da un piccolo circuito integrato (=Chip)





I progressi

I progressi ottenuti in meno di mezzo secolo potrebbero essere riassunti così:



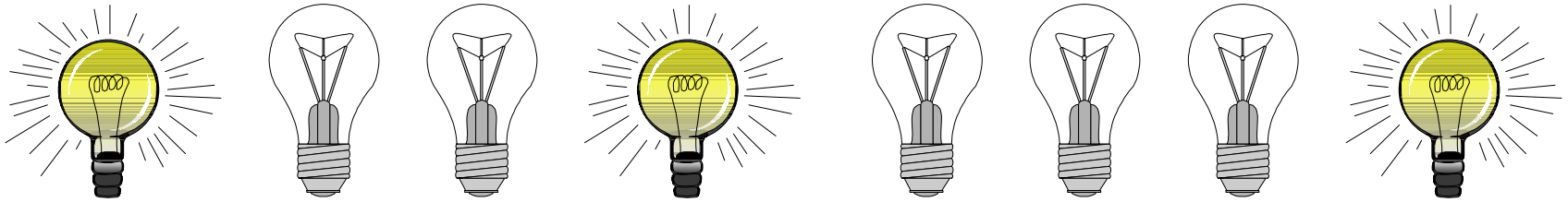
il BIT

BIT = **BI**nary digi**T** è una cifra binaria uno zero o un uno

dal latino "DIGITUS" = dito deriva
l'inglese "DIGIT" = dito e cifra

Ad esempio: 1001 0001 è un numero binario di 8 BIT
nei calcolatori viene associato alla presenza o assenza
di corrente elettrica e memorizzata così:

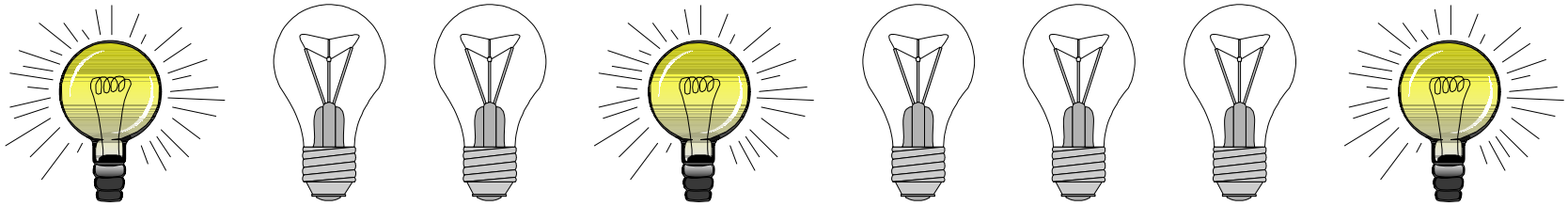
1 0 0 1 0 0 0 1



il BYTE

Il **BYTE** è un gruppo di 8 BIT = 2^3 BIT

Ad esempio 10010001 è un BYTE



I multipli del BYTE

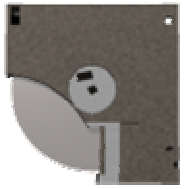
- KB = KiloByte = 2^{10} Byte = 1024 B
- MB= MegaByte = 2^{20} Byte = 1024 KB =1.048.576 B
- GB= GigaByte = 2^{30} Byte = 1024 MB=1.073.741.824 B
- TB= TeraByte = 2^{40} Byte = 1024 GB=1.099.511.627.776 B

Esempi di occupazione di memoria

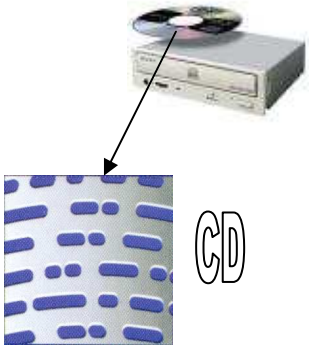
Valori esemplificativi

- La divina commedia TXT: 570 KB
- Una foto digitale JPG: 1MB
- Un minuto di musica MP3: 1MB
- Un minuto di musica HI-FI: 12MB
- Un LP HI-FI: 700MB
- Un Enciclopedia multimediale: 4GB
- Un film in DVD: tra 4 e 16GB

Memorie di massa



Floppy disk
1,44 MB



CD
700 MB

DVD
Vari standard tra 4 e 16 GB

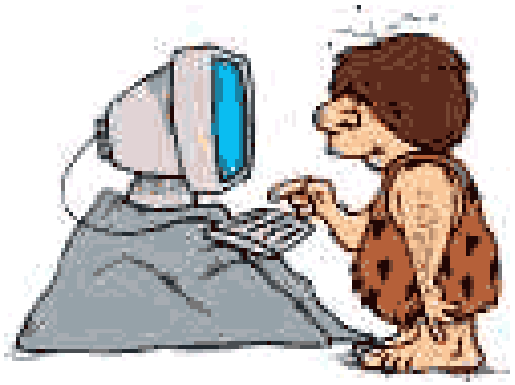
USB disk
Da 1-2-4-8-16-32…GB

Hard disk
da molti GB fino a oltre 1TB

Digitalizzare I numeri

I numeri possono essere **digitalizzati** utilizzando la matematica binaria che abbiamo visto essere equivalente a quella decimale;
il compito è complicato per gli esseri umani abituati ad usare una matematica decimale, ma semplice per i calcolatori:

0	0	
1	**1**	**$=2^0$**
10	**2**	**$=2^1$**
11	3	
100	**4**	**$=2^2$**
101	5	
110	6	
111	7	
1000	**8**	**$=2^3$**
10000	**16**	**$=2^4$**
100000	**32**	**$=2^5$**
1000000	**64**	**$=2^6$**
10000000	**128**	**$=2^7$**
100000000	**256**	**$=2^8$**
1000000000	**512**	**$=2^9$**
10000000000	**1024**	**$=2^{10}$**



Digitalizzare i caratteri alfabetici

I caratteri alfabetici sono digitalizzati utilizzando il codice ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)
una tabella che associa un numero binario di 7 bit ad un carattere,
sono così rappresentati 2^7=128 caratteri diversi

Byte	Cod.	Char
00000000	0	Null
00000001	1	Start of heading
00000010	2	Start of text
00000011	3	End of text
00000100	4	End of transmit
00000101	5	Enquiry
00000110	6	Acknowledge
00000111	7	Audible bell
00001000	8	Backspace
00001001	9	Horizontal tab
00001010	10	Line feed
00001011	11	Vertical tab
00001100	12	Form Feed
00001101	13	Carriage return
00001110	14	Shift out
00001111	15	Shift in
00010000	16	Data link escape
00010001	17	Device control 1
00010010	18	Device control 2
00010011	19	Device control 3
00010100	20	Device control 4
00010101	21	Neg. acknowledge
00010110	22	Synchronous idle
00010111	23	End trans. block
00011000	24	Cancel
00011001	25	End of medium
00011010	26	Substitution
00011011	27	Escape
00011100	28	File separator
00011101	29	Group separator
00011110	30	Record Separator
00011111	31	Unit separator

Byte	Cod.	Char
00100000	32	Spc
00100001	33	!
00100010	34	"
00100011	35	#
00100100	36	$
00100101	37	%
00100110	38	&
00100111	39	'
00101000	40	(
00101001	41	)
00101010	42	*
00101011	43	+
00101100	44	,
00101101	45	-
00101110	46	.
00101111	47	/
00110000	48	0
00110001	49	1
00110010	50	2
00110011	51	3
00110100	52	4
00110101	53	5
00110110	54	6
00110111	55	7
00111000	56	8
00111001	57	9
00111010	58	:
00111011	59	;
00111100	60	<
00111101	61	=
00111110	62	>
00111111	63	?

Byte	Cod.	Char
01000000	64	@
01000001	65	A
01000010	66	B
01000011	67	C
01000100	68	D
01000101	69	E
01000110	70	F
01000111	71	G
01001000	72	H
01001001	73	I
01001010	74	J
01001011	75	K
01001100	76	L
01001101	77	M
01001110	78	N
01001111	79	O
01010000	80	P
01010001	81	Q
01010010	82	R
01010011	83	S
01010100	84	T
01010101	85	U
01010110	86	V
01010111	87	W
01011000	88	X
01011001	89	Y
01011010	90	Z
01011011	91	[
01011100	92	\
01011101	93	]
01011110	94	^
01011111	95	_

Byte	Cod.	Char
01100000	96	`
01100001	97	a
01100010	98	b
01100011	99	c
01100100	100	d
01100101	101	e
01100110	102	f
01100111	103	g
01101000	104	h
01101001	105	i
01101010	106	j
01101011	107	k
01101100	108	l
01101101	109	m
01101110	110	n
01101111	111	o
01110000	112	p
01110001	113	q
01110010	114	r
01110011	115	s
01110100	116	t
01110101	117	u
01110110	118	v
01110111	119	w
01111000	120	x
01111001	121	y
01111010	122	z
01111011	123	{
01111100	124	|
01111101	125	}
01111110	126	~
01111111	127	Del

Byte	Cod.	Char
01000000	64	@
01000001	65	A
01000010	66	B
01000011	67	C
01000100	68	D
01000101	69	E
01000110	70	F

Byte	Cod.	Char
01100000	96	`
01100001	97	a
01100010	98	b
01100011	99	c
01100100	100	d
01100101	101	e
01100110	102	f

Digitalizzare i caratteri alfabetici

Il codice ASCII si è poi **esteso** a 8 bit per aggiungere altri caratteri in particolare le lettere accentate in uso nel mondo occidentale così oggi il codice ASCI comprende 256 caratteri diversi (2^8=256)

Byte	Cod.	Char
10000000	128	Ç
10000001	129	ü
10000010	130	é
10000011	131	â
10000100	132	ä
10000101	133	à
10000110	134	å
10000111	135	ç
10001000	136	ê
10001001	137	ë
10001010	138	è
10001011	139	ï
10001100	140	î
10001101	141	ì
10001110	142	Ä
10001111	143	Å
10010000	144	É
10010001	145	æ
10010010	146	Æ
10010011	147	ô
10010100	148	ö
10010101	149	ò
10010110	150	û
10010111	151	ù
10011000	152	ÿ
10011001	153	Ö
10011010	154	Ü
10011011	155	ø
10011100	156	£
10011101	157	Ø
10011110	158	×
10011111	159	ƒ
10100000	160	á
10100001	161	í
10100010	162	ó
10100011	163	ú
10100100	164	ñ
10100101	165	Ñ
10100110	166	ª
10100111	167	º
10101000	168	¿
10101001	169	®
10101010	170	¬
10101011	171	½
10101100	172	¼
10101101	173	¡
10101110	174	«
10101111	175	»
10110000	176	▒
10110001	177	▒
10110010	178	▒
10110011	179	┤
10110100	180	┤
10110101	181	Á
10110110	182	Â
10110111	183	À
10111000	184	©
10111001	185	╣
10111010	186	║
10111011	187	╗
10111100	188	╝
10111101	189	¢
10111110	190	¥
10111111	191	┐
11000000	192	└
11000001	193	┴
11000010	194	┬
11000011	195	├
11000100	196	─
11000101	197	┼
11000110	198	ã
11000111	199	Ã
11001000	200	╚
11001001	201	╔
11001010	202	╩
11001011	203	╦
11001100	204	╠
11001101	205	═
11001110	206	╬
11001111	207	¤
11010000	208	ð
11010001	209	Ð
11010010	210	Ê
11010011	211	Ë
11010100	212	È
11010101	213	ı
11010110	214	Í
11010111	215	Î
11011000	216	Ï
11011001	217	┘
11011010	218	┌
11011011	219	█
11011100	220	▄
11011101	221	¦
11011110	222	Ì
11011111	223	▀
11100000	224	Ó
11100001	225	ß
11100010	226	Ô
11100011	227	Ò
11100100	228	õ
11100101	229	Õ
11100110	230	µ
11100111	231	þ
11101000	232	Þ
11101001	233	Ú
11101010	234	Û
11101011	235	Ù
11101100	236	ý
11101101	237	Ý
11101110	238	¯
11101111	239	´
11110000	240	
11110001	241	±
11110010	242	‗
11110011	243	¾
11110100	244	¶
11110101	245	§
11110110	246	÷
11110111	247	¸
11111000	248	°
11111001	249	¨
11111010	250	·
11111011	251	¹
11111100	252	³
11111101	253	²
11111110	254	■
11111111	255	[ ]

Byte	Cod.	Char
10000000	128	Ç
10000001	129	ü
10000010	130	é
10000011	131	â
10000100	132	ä
10000101	133	à
10000110	134	å

Byte	Cod.	Char
10100000	160	á
10100001	161	í
10100010	162	ó
10100011	163	ú
10100100	164	ñ
10100101	165	Ñ
10100110	166	ª

I sistemi operativi riconoscono i file in codice ASCII marcandoli con l'estensione .TXT (in Windows il "blocco note" salva in formato TXT, ma anche con Word è possibile salvare in questo formato)

Digitalizzare immagini in bianco e nero

Le immagini sono digitalizzate utilizzando “mappe di BIT”: ossia suddividendo l’immagine in piccoli punti ed associando i punti neri al numero 0 e i punti bianchi al numero 1.
In questo modo ogni punto (detto PIXEL contrazione dell’inglese PICture Element) occuperà esattamente un BIT.

I mosaici romani sono un esempio di immagine “Digitale” i cui PIXEL corrispondono alle tessere bianche o nere

```
00000000000111100011000000001100000
00000000000111100011000011111111111
11111111111111100011000101011101001
11111111111111100011000010101010101
00000000000000000011000010101010111
11111100011001111100000000000110000
00011111111111111111111000011101010
11100011101011101001010101010110101
01100101010101010101010101010101011
01010011010101010111111111000110011
11100000000000110000000111111111111
11111111000011101010111000111010111
01001010101010110101011001010101010
10101010101010101010110101001101010
```

Le immagini a colori

Anche le immagini a colori sono digitalizzate utilizzando “mappe di BIT”, ma saranno necessari più BIT per ogni pixel per rappresentare più colori:

Ad esempio un’immagine a 16 colori avrà bisogno di 4 BIT per ogni PIXEL (2^4=16):

```
Bianco  =1111        Verde1  =0111
Grigio1 =1110        Verde2  =0110
Grigio2 =1101        Giallo  =0101
Rosso   =1100        Celeste =0100
Rosa    =1011        Azzurro =0011
Viola   =1010        Blu1    =0010
Lilla   =1001        Blu2    =0001
Arancio =1000        Nero    =0000
```

Simili tabelle di associazione dei colori sono possibili per immagini a più colori:
256 colori (8 BIT per ogni pixel) 2^8=256
16 milioni di colori (24 BIT per ogni pixel) 2^{24}=16.667.216

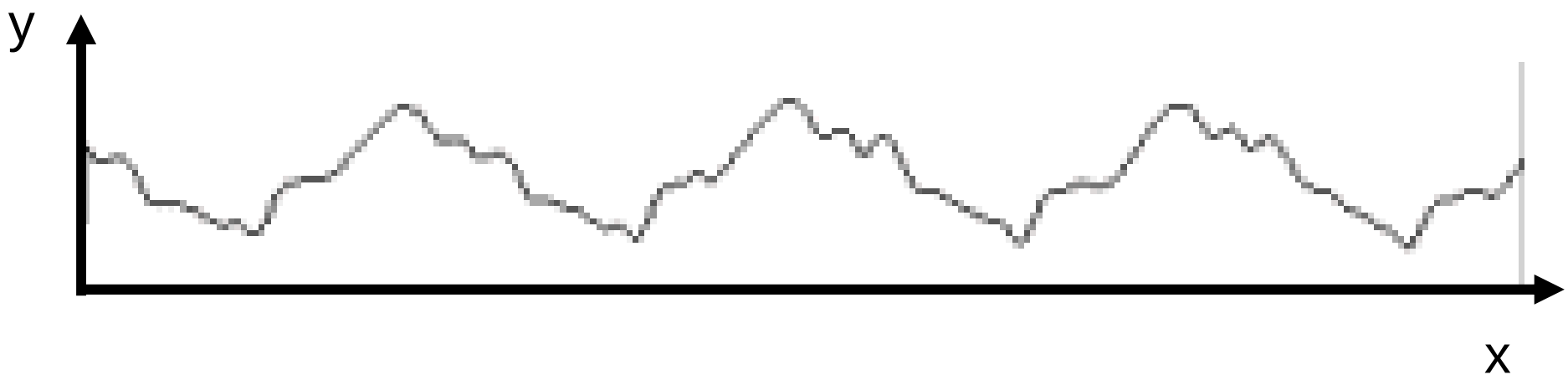
per memorizzare un’immagine di 800x600 pixel a 24 bit saranno necessari 11.520.000 bit = 1.440.000 Byte

I sistemi operativi riconoscono i file immagine a mappa di bit marcandoli con l’estensione .BMP
(in Windows il “paint” salva in formato BMP, ma anche con altre applicazioni è possibile salvare in questo formato)

Digitalizzare i suoni

È possibile digitalizzare i suoni trasformandoli in funzioni.

Se queste funzioni sono espresse con la matematica binaria il computer le può gestire…



per memorizzare un minuto di musica in qualità HI-FI occorrono circa 11MB,
per memorizzare un LP di circa 70 minuti occorre un intero CD da 700MB

I sistemi operativi riconoscono i file musicali in qualità HI-FI marcandoli con l'estensione .WAV e quelli di qualità ridotta con l'estensione .MP3

Digitalizzare i filmati

I filmati sono una successione di immagini (fotogrammi), accompagnate da una colonna sonora.

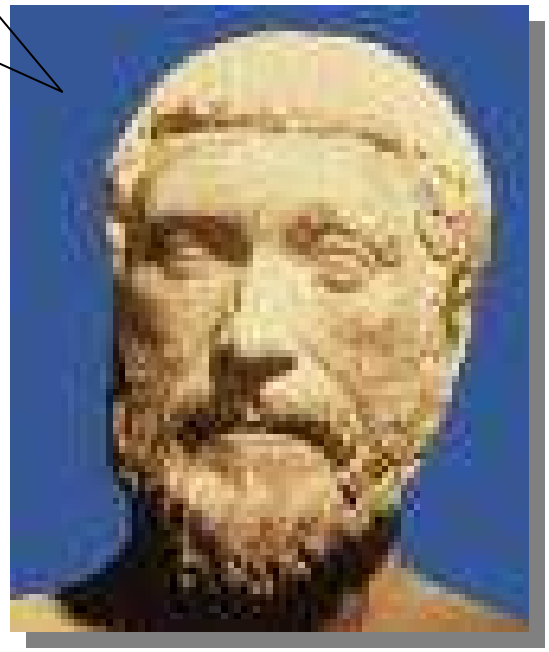
Digitalizzando, uno per uno, prima tutti i fotogrammi (ogni fotogramma è un'immagine) e poi la colonna sonora, si digitalizza un intero film.

per memorizzare un film di due ore è necessario un intero DVD della capacità di alcuni GB

I sistemi operativi riconoscono i file video in qualità DVD marcandoli con l'estensione .VOB e quelli di qualità ridotta in vari modi con le estensione .MPG .AVI .WMV .MOV ecc…

Tutto è numero!

Esclamò Pitagora di Samo nel sesto secolo avanti Cristo…



…ora abbiamo un motivo in più per dargli ragione!