

# Memorizzazione e rappresentazione di dati

Eugenio G. Omodeo



UNIVERSITÀ  
DEGLI STUDI DI TRIESTE

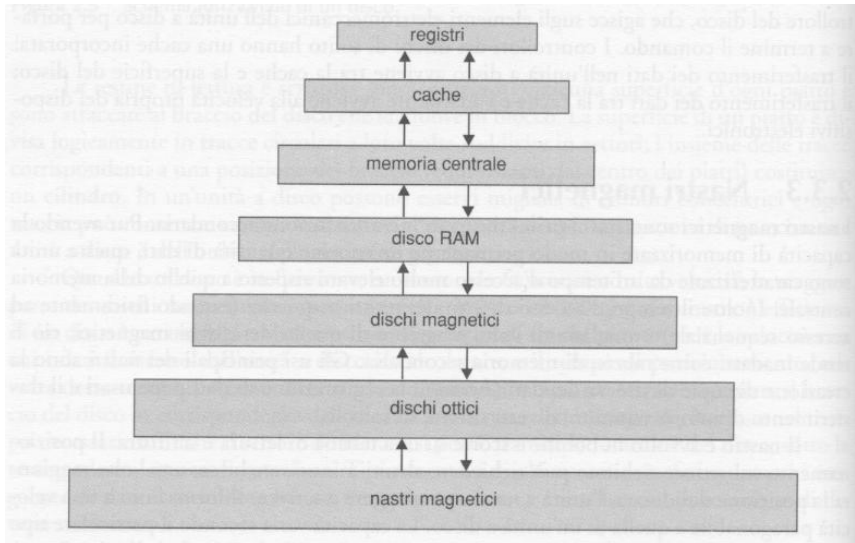
Dip. Matematica e Geoscienze — DMI



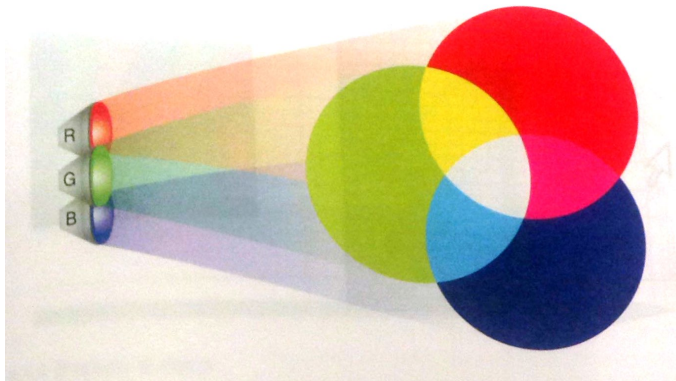
UNIVERSITÀ  
DEGLI STUDI DI TRIESTE

Trieste, 06/12/2022

# GERARCHIA DELLE MEMORIE IN UN COMPUTER



# ESEMPI DI COSA SERVE MEMORIZZARE: COLORI



**Figura 6.1** Combinazione della luce RGB a piena intensità per creare i diversi colori.



# ESEMPI DI COSA SERVE MEMORIZZARE: CARATTERI

ASCII	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>	
0001	P <sub>L</sub>	P <sub>I</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	N <sub>K</sub>	S <sub>V</sub>	E <sub>Z</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>	
0010		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	P <sub>T</sub>	
1000	S <sub>O</sub>	S <sub>I</sub>	S <sub>2</sub>	S <sub>3</sub>	I <sub>N</sub>	N <sub>L</sub>	S <sub>S</sub>	E <sub>S</sub>	H <sub>S</sub>	H <sub>J</sub>	Y <sub>S</sub>	P <sub>D</sub>	P <sub>V</sub>	R <sub>I</sub>	S <sub>2</sub>	S <sub>3</sub>	
1001	P <sub>C</sub>	P <sub>I</sub>	P <sub>Z</sub>	S <sub>E</sub>	C <sub>C</sub>	M <sub>M</sub>	S <sub>P</sub>	E <sub>P</sub>	O <sub>S</sub>	O <sub>O</sub>	O <sub>A</sub>	C <sub>S</sub>	S <sub>T</sub>	O <sub>S</sub>	P <sub>M</sub>	A <sub>P</sub>	
1010	°	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯	
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿	
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	
1101	Ð	Ñ	Ò	Ó	Ô	Õ	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß		
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	
1111	ø	ñ	ò	ó	ô	õ	÷	ø	ù	ú	û	ü	ý	þ	ÿ		



# ESEMPI DI COSA SERVE MEMORIZZARE: CARATTERI

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	&#032;	Space	64	40	100	&#064;	@	96	60	140	&#096;	`
1	1	001	Start of Header	33	21	041	&#033;	!	65	41	101	&#065;	A	97	61	141	&#097;	a
2	2	002	Start of Text	34	22	042	&#034;	"	66	42	102	&#066;	B	98	62	142	&#098;	b
3	3	003	End of Text	35	23	043	&#035;	#	67	43	103	&#067;	C	99	63	143	&#099;	c
4	4	004	End of Transmission	36	24	044	&#036;	\$	68	44	104	&#068;	D	100	64	144	&#100;	d
5	5	005	Enquiry	37	25	045	&#037;	%	69	45	105	&#069;	E	101	65	145	&#101;	e
6	6	006	Acknowledgment	38	26	046	&#038;	&	70	46	106	&#070;	F	102	66	146	&#102;	f
7	7	007	Bell	39	27	047	&#039;	'	71	47	107	&#071;	G	103	67	147	&#103;	g
8	8	010	Backspace	40	28	050	&#040;	(	72	48	110	&#072;	H	104	68	150	&#104;	h
9	9	011	Horizontal Tab	41	29	051	&#041;	)	73	49	111	&#073;	I	105	69	151	&#105;	i
10	A	012	Line feed	42	2A	052	&#042;	*	74	4A	112	&#074;	J	106	6A	152	&#106;	j
11	B	013	Vertical Tab	43	2B	053	&#043;	+	75	4B	113	&#075;	K	107	6B	153	&#107;	k
12	C	014	Form feed	44	2C	054	&#044;	,	76	4C	114	&#076;	L	108	6C	154	&#108;	l
13	D	015	Carriage return	45	2D	055	&#045;	-	77	4D	115	&#077;	M	109	6D	155	&#109;	m
14	E	016	Shift Out	46	2E	056	&#046;	.	78	4E	116	&#078;	N	110	6E	156	&#110;	n
15	F	017	Shift In	47	2F	057	&#047;	/	79	4F	117	&#079;	O	111	6F	157	&#111;	o
16	10	020	Data Link Escape	48	30	060	&#048;	0	80	50	120	&#080;	P	112	70	160	&#112;	p
17	11	021	Device Control 1	49	31	061	&#049;	1	81	51	121	&#081;	Q	113	71	161	&#113;	q
18	12	022	Device Control 2	50	32	062	&#050;	2	82	52	122	&#082;	R	114	72	162	&#114;	r
19	13	023	Device Control 3	51	33	063	&#051;	3	83	53	123	&#083;	S	115	73	163	&#115;	s
20	14	024	Device Control 4	52	34	064	&#052;	4	84	54	124	&#084;	T	116	74	164	&#116;	t
21	15	025	Negative Ack.	53	35	065	&#053;	5	85	55	125	&#085;	U	117	75	165	&#117;	u
22	16	026	Synchronous idle	54	36	066	&#054;	6	86	56	126	&#086;	V	118	76	166	&#118;	v
23	17	027	End of Trans. Block	55	37	067	&#055;	7	87	57	127	&#087;	W	119	77	167	&#119;	w
24	18	030	Cancel	56	38	070	&#056;	8	88	58	130	&#088;	X	120	78	170	&#120;	x
25	19	031	End of Medium	57	39	071	&#057;	9	89	59	131	&#089;	Y	121	79	171	&#121;	y
26	1A	032	Substitute	58	3A	072	&#058;	:	90	5A	132	&#090;	Z	122	7A	172	&#122;	z
27	1B	033	Escape	59	3B	073	&#059;	;	91	5B	133	&#091;	[	123	7B	173	&#123;	{
28	1C	034	File Separator	60	3C	074	&#060;	<	92	5C	134	&#092;	\	124	7C	174	&#124;	
29	1D	035	Group Separator	61	3D	075	&#061;	=	93	5D	135	&#093;	]	125	7D	175	&#125;	}
30	1E	036	Record Separator	62	3E	076	&#062;	>	94	5E	136	&#094;	^	126	7E	176	&#126;	~
31	1F	037	Unit Separator	63	3F	077	&#063;	?	95	5F	137	&#095;	_	127	7F	177	&#127;	Del

# ESEMPI DI COSA SERVE MEMORIZZARE: PRODOTTI

8 → 1001000 →



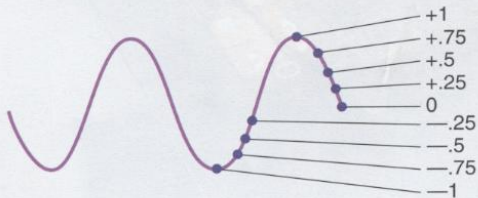
Codice produttore

Codice prodotto

UNIVERSITÀ  
GLI STUDI DI TRIESTE

## *Digitale o analogico?*

I **segnali analogici** sono paragonabili a un'onda e trasmettono un ampio spettro di informazioni. Il nostro è un mondo analogico. Il corpo umano è programmato per percepire e interpretare onde luminose analogiche: noi non percepiamo soltanto i punti massimi e minimi di queste onde, ma anche quelli intermedi. Quando parliamo, il messaggio viene trasportato da onde analogiche e lo stesso accade quando ascoltiamo della musica. Ogni parte dell'onda trasporta informazioni.

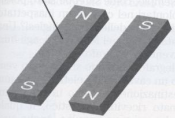


*Segnali analogici*

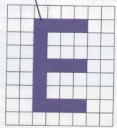


# ANALOGICO VS DIGITALE

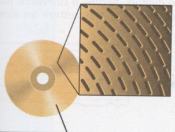
Le particelle magnetiche presenti sulla superficie di un disco o di un nastro possono essere distribuite in modo da puntare in direzioni opposte, una rappresentante la cifra 1, l'altra la 0 (polarità).



Su uno schermo i punti possono essere accesi, quando corrispondono a 1, o spenti, quando corrispondono a 0.



I misuratori di tensione possono indicare alta tensione, pari a 1, o bassa tensione, pari a 0.

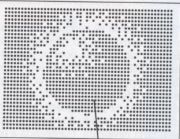


Le scanalature sulla superficie di un CD-ROM o di altri dispositivi ottici di memorizzazione riflettono la luce del laser in modo diverso rispetto alle zone non scanalate: le prime corrispondono a 1, le seconde a 0.

I dispositivi a due posizioni, come i transistor, possono rappresentare 1 quando sono accesi e 0 quando sono spenti.



I puntini vengono stampati quando corrispondono a 1 e non stampati quando corrispondono a 0. Tenendo quest'immagine a una certa distanza, potete vedere la figura che forma.

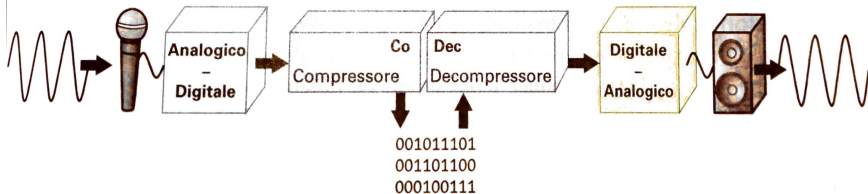


Dispositivi digitali



**ADC-CoDec-DAC.** Il processo di digitalizzazione opera come segue: il suono viene rilevato da un microfono (detto trasduttore perché converte l'onda sonora in un'onda elettrica), quindi il segnale elettrico viene passato a un **convertitore analogico-digitale** (*analog-to-digital converter*, **ADC**), che campiona l'onda continua a intervalli regolari e per ogni campione produce numeri binari. Questi numeri vengono poi compressi (vedi Figura 6.14) e salvati in memoria.

Per riprodurre il suono avviene il procedimento inverso: i numeri in memoria vengono letti dal decompressore, che produce numeri binari adatti alla riproduzione. Questi vengono passati a un **convertitore digitale-analogico** (*digital-to-analog converter*, **DAC**), che crea un'onda elettrica interpolando i valori digitali, ossia riempiendo gli spazi vuoti o passando gradualmente da un valore all'altro. Il segnale elettrico viene poi passato a un altoparlante che converte l'onda elettrica in onda sonora, come mostrato nella Figura 6.14.



*“Un **bit** è il dato prodotto in una singola osservazione Panda”<sup>1</sup> [SHRA20, p. 127]*

*“[...] Perché **byte** si scrive con la **y** ? Per capire la risposta, dovete prima di tutto sapere che la memoria dei computer è soggetta a errori ( uno zero che diventa un uno e viceversa ). Le cause possono essere diverse, e arrivano fino all’influenza dei raggi cosmici. Sul serio. La possibilità che si verifichino errori, poi, cresce quando si inviano i bit su una rete di comunicazione.” [SHRA20, p. 139]*

---

<sup>1</sup>Da *Presence and Absence*

## Informazioni in ingresso

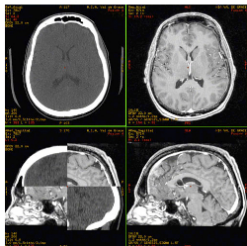
A,a,B,b,... simboli da tastiera



suoni

$12+37 (3 -5) =$

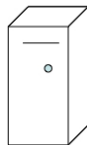
numeri



immagini

## Calcolatore

001011011101



Codifica

Numeri  $\begin{matrix} \leftarrow \\ \rightarrow \end{matrix}$  numerali



Numeri decimali	Numeri binari			
0.....				0
1.....				1
2.....			1	0
3.....			1	1
4.....		1	0	0
5.....		1	0	1
6.....		1	1	0
7.....		1	1	1
8.....	1	0	0	0
9.....	1	0	0	1
10.....	1	0	1	0

*Numeri decimali e binari*



# COSA INTENDIAMO PER NUMERALE ?

## DEFINIZIONE:

Per *numera*le intendiamo una sequenza ( finita ) di cifre.

Le *cifre* sono particolari caratteri — quali ?



# COSA INTENDIAMO PER NUMERALE ?

## DEFINIZIONE:

Per *numerale* intendiamo una sequenza ( finita ) di cifre.

Le *cifre* sono particolari caratteri — quali ?

## CONVENZIONI:

- 1 Quando numeriamo in una base  $b$  tale che  $1 < b \leq 10$  utilizziamo come cifre i caratteri '0', '1'..., fino a  $b$  escluso.



# COSA INTENDIAMO PER NUMERALE ?

## DEFINIZIONE:

Per *numerale* intendiamo una sequenza ( finita ) di cifre.

Le *cifre* sono particolari caratteri — quali ?

## CONVENZIONI:

- 1 Quando numeriamo in una base  $b$  tale che  $1 < b \leq 10$  utilizziamo come cifre i caratteri '0', '1'..., fino a  $b$  escluso.
- 2 Quando vogliamo andar oltre, utilizzeremo come 'cifre' le lettere consecutive 'a', 'b',... fin dove occorre.





# COSA INTENDIAMO PER NUMERALE ?

## DEFINIZIONE:

Per *numerales* intendiamo una sequenza ( finita ) di cifre.

Le *cifre* sono particolari caratteri — quali ?

## CONVENZIONI:

- 1 Quando numeriamo in una base  $b$  tale che  $1 < b \leq 10$  utilizziamo come cifre i caratteri '0', '1'..., fino a  $b$  escluso.
- 2 Quando vogliamo andar oltre, utilizzeremo come 'cifre' le lettere consecutive 'a', 'b',... fin dove occorre.

**Esempio.** Le cifre del sistema **esadecimale** ( i.e., in base 16 ) sono:

0 1 2 3 4 5 6 7 8 9 a b c d e f



## ESEMPI:

101 rappresenta:

- in base 10, il numero **centouno**
- in base 2,
- in base 3,



## ESEMPI:

101 rappresenta:

- in base 10, il numero **centouno**
- in base 2, il numero **cinque**
- in base 3,



## ESEMPI:

101 rappresenta:

- in base 10, il numero **centuno**
- in base 2, il numero **cinque**
- in base 3, il numero **dieci**



## ESEMPI:

$10a$  rappresenta:

- in base 16,
- in base 12,
- in base 8,



## ESEMPI:

$10a$  rappresenta:

- in base 16, il numero duecentosessantasei
- in base 12,
- in base 8,



## ESEMPI:

$10a$  rappresenta:

- in base 16,
- in base 12, il numero **centocinquantaquattro**
- in base 8,



## ESEMPI:

$10_a$  rappresenta:

- in base 16,
- in base 12,
- in base 8, nessun numero





## ESEMPI:

10a rappresenta:

- in base 16, il numero duecentosessantasei
- in base 12, il numero centocinquantaquattro
- in base 8, nessun numero

In generale, se per  $i = 0, \dots, n$ :

- ciascuna  $k_i$  è una cifra e
- $c_i$  è il valore corrispondente a  $k_i$

allora, rispetto a una base  $b > \max\{c_0, c_1, \dots, c_n\}$ ,

il numerale  $k_n \cdots k_1 k_0$  rappresenta il numero

$$c_n b^n + c_{n-1} b^{n-1} + \cdots + c_1 b + c_0$$



## L'IDEA:

- 0 sia inizialmente  $q$  il numero dato
- 1 effettuare reiteratamente la divisione intera di  $q$  per la base  $b$ , rimpiazzando ogni volta il valore di  $q$  con il *quoziente* — conservare ogni *resto*



## L'IDEA:

- 0 sia inizialmente  $q$  il numero dato
- 1 effettuare reiteratamente la divisione intera di  $q$  per la base  $b$ , rimpiazzando ogni volta il valore di  $q$  con il *quoziente* — conservare ogni *resto*
- 2 terminare il ciclo quando  $q$  diventa 0



## L'IDEA:

- 0 sia inizialmente  $q$  il numero dato
- 1 effettuare reiteratamente la divisione intera di  $q$  per la base  $b$ , rimpiazzando ogni volta il valore di  $q$  con il *quoziente* — conservare ogni *resto*
- 2 terminare il ciclo quando  $q$  diventa 0
- 3 raccogliere i resti di sotto in su, convertendo ciascuno nella cifra corrispondente



## L'IDEA:

- 0 sia inizialmente  $q$  il numero dato
- 1 effettuare reiteratamente la divisione intera di  $q$  per la base  $b$ , rimpiazzando ogni volta il valore di  $q$  con il *quoziente* — conservare ogni *resto*
- 2 terminare il ciclo quando  $q$  diventa 0
- 3 raccogliere i resti di sotto in su, convertendo ciascuno nella cifra corrispondente

**Esempio:**  $371/2 = 185$  con resto  $1$ ;  $185/2 = 92$  con resto  $1$ ;  
 $92/2 = 46$  con r.  $0$ ;  $46/2 = 23$  con r.  $0$ ;  $23/2 = 11$  con r.  $1$ ;  
 $11/2 = 5$  con r.  $1$ ;  $5/2 = 2$  con r.  $1$ ;  $2/2 = 1$  con resto  $0$ ;  
 $1/2 = 0$  con r.  $1$

"101110011" è 371 espresso in base 2



# LA CONVERS. DI CUI SOPRA, IN MATLAB

Dammi un valore intero: 347

fg >>

Live Editor - /Users/eugenioomodeo/Documents/MATLAB/Introduzione\_Matlab\_1.mlx

intro\_if.m x pari\_dispari.m x vettore.m x conversBase.m x Introduzione\_Matlab\_1 (1).mlx x Introduzione\_Matlab\_1.mlx x Introduzione\_Matlab\_1 (2).mlx x testaOcroce.m x p

Conversione di base: Esprimi un numero in diverse basi:

```
67 clear, clc
68
69 cifre=['0','1','2','3','4','5','6','7','8','9','a','b','c','d','e','f'];
70
71 N=input('Dammi un valore intero: ');
72
73 if N < 0
74
75     fprintf('%1.0f e un numero negativo',N);
76
77 else
78
79     for base = [2 , 3, 8, 16]
80
81         fprintf('Sto convertendo il tuo numero, %1.0f, alla base %1.0f',N,base);
82
83         Ncopia = N; % per non rovinare l'originale
84
85         NinBase = ''; % diventera' il numero trasformato
86
87         while Ncopia ~= 0 % qui inizia la conversione in base `base`
88
89             NinBase = strcat( cifre(rem(Ncopia,base)+1), NinBase )
90             % NotaBene: rem designa l'operazione `Remainder after division`.
91
92             Ncopia = floor(Ncopia / base);
93
94         end
95
96         if size( NinBase ) == 0
97
98             NinBase = '0';
99
100         end
101
102         fprintf('Ecco il numero %1.0f espresso in base %1.0f: ', N, base), disp(NinBase);
103
104     end
105
106 end
```

```
Sto convertendo il tuo numero, 347, alla base 2
NinBase = '1'
NinBase = '11'
NinBase = '011'
NinBase = '1011'
NinBase = '11011'
NinBase = '011011'
NinBase = '1011011'
NinBase = '01011011'
NinBase = '101011011'
Ecco il numero 347 espresso in base 2: 101011011
Sto convertendo il tuo numero, 347, alla base 3
NinBase = '2'
NinBase = '12'
NinBase = '212'
NinBase = '0212'
NinBase = '10212'
NinBase = '110212'
Ecco il numero 347 espresso in base 3: 110212
Sto convertendo il tuo numero, 347, alla base 8
NinBase = '3'
NinBase = '33'
NinBase = '533'
Ecco il numero 347 espresso in base 8: 533
Sto convertendo il tuo numero, 347, alla base 16
```

Quanta informazione posso  
codificare in  $n$  bit ?



Quanta informazione posso  
codificare in  $n$  bit ?

situazione idealizzata. . .





# CONFIGURAZ. POSSIBILI PER UNA SEQUENZA DI BITS

CON  $n$  BIT...

... posso rappresentare informazioni distinte.



# CONFIGURAZ. POSSIBILI PER UNA SEQUENZA DI BITS

CON  $n$  BIT...

... posso rappresentare  $2^n$  informazioni distinte.



# CONFIGURAZ. POSSIBILI PER UNA SEQUENZA DI BITS

CON  $n$  BIT...

... posso rappresentare  $2^n$  informazioni distinte.

ESEMPI:

- 1 con 8 bit, ho a disposizione 256 'simboli' ( alfabeto adeguato a moltissime situazioni )

RIESTE

CON  $n$  BIT...

... posso rappresentare  $2^n$  informazioni distinte.

ESEMPI:

- 1 con 8 bit, ho a disposizione 256 'simboli' ( alfabeto adeguato a moltissime situazioni )
- 2 con 10 bit, circa 1\_000; in effetti,

$$2^{10} = 1024 = 1Kbit$$

# CONFIGURAZ. POSSIBILI PER UNA SEQUENZA DI BITS

CON  $n$  BIT...

... posso rappresentare  $2^n$  informazioni distinte.

ESEMPI:

① con 8 bit, ho a disposizione 256 'simboli' ( alfabeto adeguato a moltissime situazioni )

② con 10 bit, circa 1\_000; in effetti,

$$2^{10} = 1024 = 1Kbit$$

③ con 20 bit, circa 1\_000\_000; in effetti,

$$2^{20} = 1024^2 = 1Mbit$$

- ① **ASCII** ( acronimo di 'American Standard Code for Information Interchange' ), del 1968, oggi variamente esteso da 7 a 8 bit.



- ① **ASCII** ( acronimo di 'American Standard Code for Information Interchange' ), del 1968, oggi variamente esteso da 7 a 8 bit.
- ② **UNICODE** ( sostenuto dal W3C consortium ), oggi variamente esteso da 16 a 21 bit.



- ① **ASCII** ( acronimo di 'American Standard Code for Information Interchange' ), del 1968, oggi variamente esteso da 7 a 8 bit.
- ② **UNICODE** ( sostenuto dal W3C consortium ), oggi variamente esteso da 16 a 21 bit.

## ESERCIZI:

- ① quanti colori posso rappresentare con 6 cifre esadecimali ?





- ① **ASCII** ( acronimo di 'American Standard Code for Information Interchange' ), del 1968, oggi variamente esteso da 7 a 8 bit.
- ② **UNICODE** ( sostenuto dal W3C consortium ), oggi variamente esteso da 16 a 21 bit.

## ESERCIZI:

- ① quanti colori posso rappresentare con 6 cifre esadecimali ?
- ② quanti caratteri posso rappresentare con UTF-16 ?



## I Numeri degli Encoding

Ecco alcuni esempi per capire la potenza di vari Encoding

### *UTF-8*

UTF-8 può occupare spazi diversi per memorizzare caratteri diversi, in particolare:

- Usa 1 byte per memorizzare i caratteri ASCII;
- Usa 2 byte per memorizzare altri 1920 caratteri (Romanico, Greco, Cirillico, etc);
- Usa 3 byte per memorizzare 63488 caratteri (tra cui cinese e giapponese);
- Usa 4-5-6 byte per altri 2147418112 non tutti utilizzati.

### *UTF-16*

UTF-16 memorizza tutti i primi 65536 caratteri in 2 byte, gli altri in 4 byte.

### *UCS-2*

UCS-2 rappresenta solo i primi 65536 caratteri Unicode e occupa sempre 2 byte.

### *UCS-4*

Tutti i caratteri sono rappresentati a 4 byte e per ora tutti sono rappresentabili.





Lawrence Snyder, Laura Henry Ray, Alessandro Amoroso.  
*FLUENCY – Conoscere e usare l'informatica.*  
Pearson, MyLab. 7<sup>a</sup> edizione, 2020.

