

# Architettura degli Elaboratori

## Esercitazione



**Barbara Masucci**

UNIVERSITA' DEGLI STUDI DI SALERNO

**DIPARTIMENTO DI INFORMATICA**

**DIPARTIMENTO DI ECCELLENZA**

# Su cosa ci esercitiamo oggi?

- Notazione posizionale pesata per numeri **interi**
  - Rappresentazione nelle basi 10, 2, 8, 16
  - Conversioni tra le diverse basi
- Notazione posizionale pesata per **frazioni proprie**
  - Rappresentazione in base 2
  - Conversioni Decimale-Binario e Binario-Decimale
- Aritmetica in binario



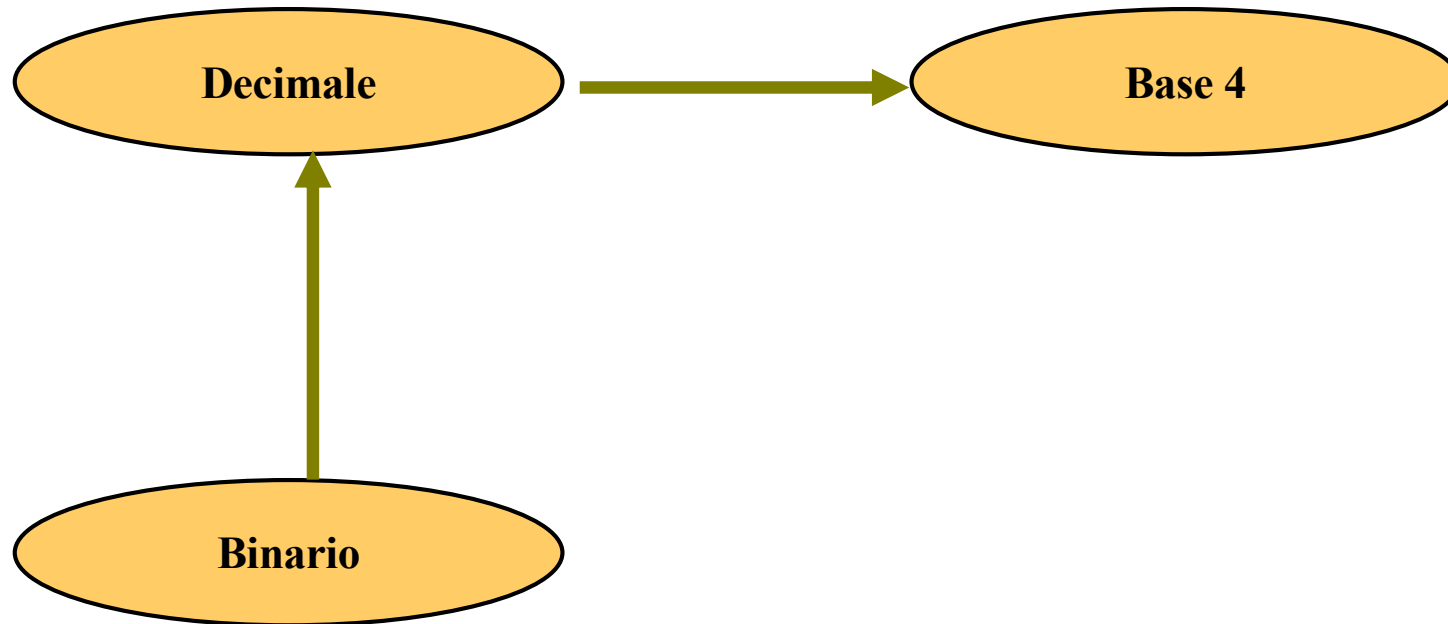
# Esercizio 1

- Descrivere un algoritmo che dato un numero  $N=(b_7 b_6 \dots b_0)_2$ , ne fornisca la **rappresentazione in base 4**
- L'algoritmo non deve utilizzare conversioni intermedie
- E' necessario giustificare perché l'algoritmo è corretto (funziona sempre)
- Eseguire l'algoritmo proposto sulla sequenza  $(01000110)_2$



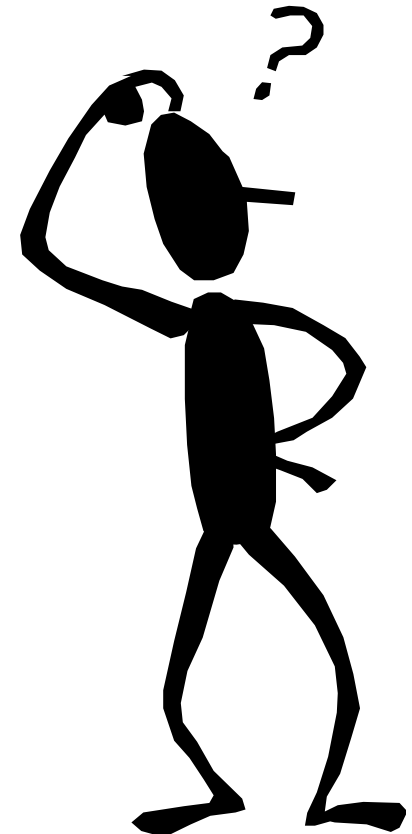
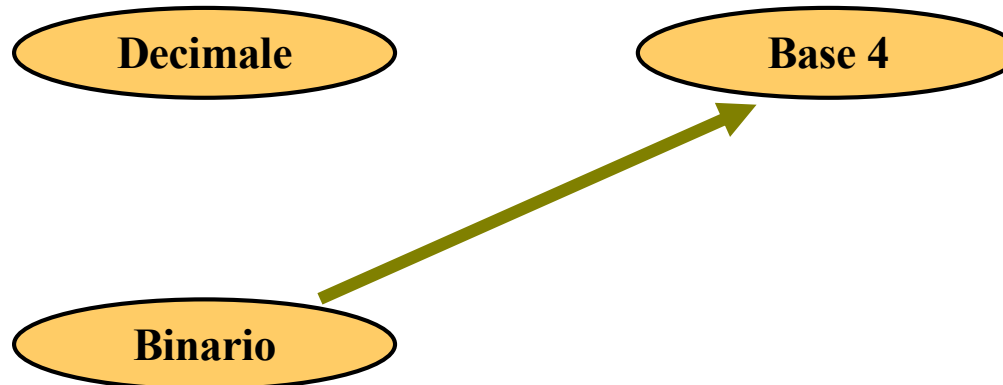
# Esercizio 1: Soluzione

- Possiamo convertire un numero da Binario a Base 4 facendo **due conversioni**, **ma non dobbiamo farlo!**



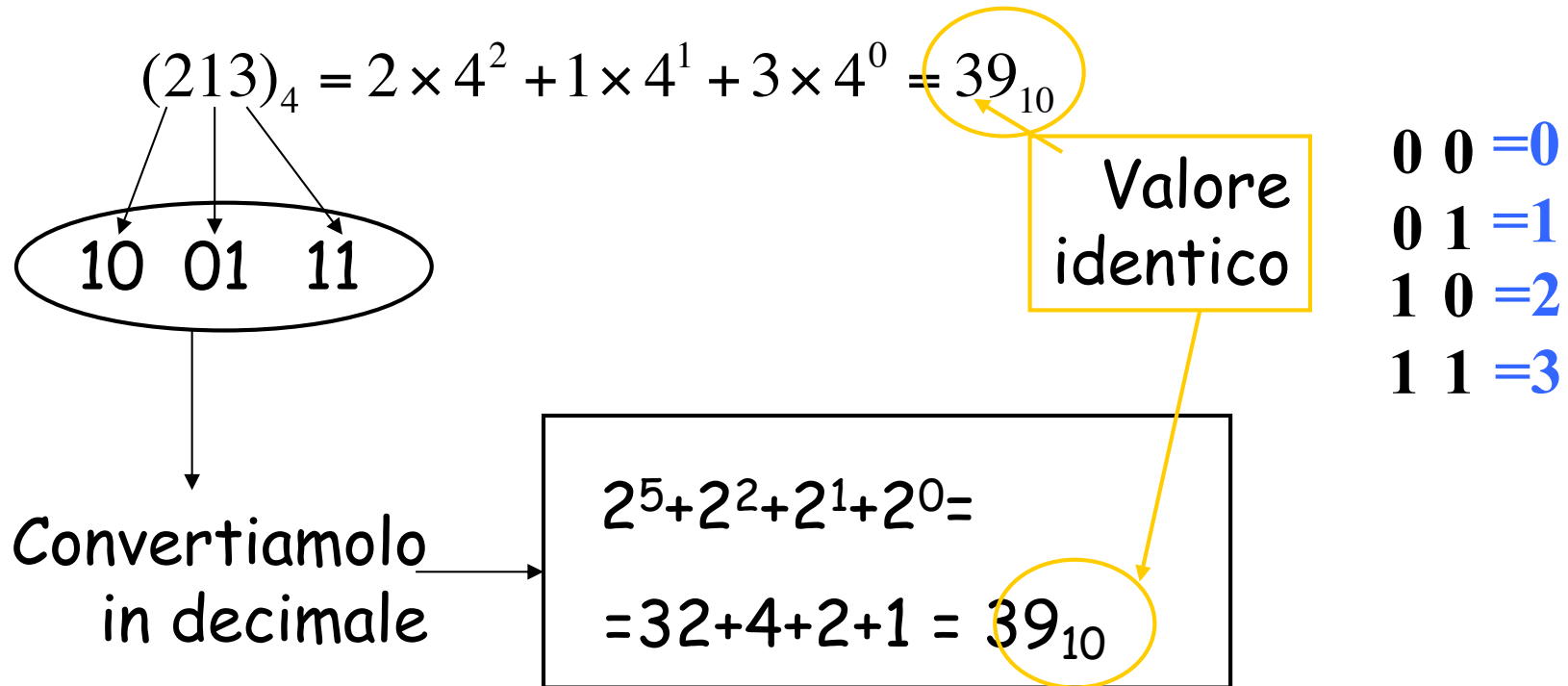
# Esercizio 1: Soluzione

Esiste un metodo per passare dalla  
rappresentazione binaria a quella  
in base 4 senza passare  
per quella decimale?



# Esercizio 1: Soluzione

Rappresentazione in base 4



# Esercizio 1: Soluzione

$$N = (b_7 b_6 \dots b_1 b_0)_2$$

$$= b_7 2^7 + b_6 2^6 + \dots + b_1 2^1 + b_0$$

$$= b_7 \times 2^7 + b_6 \times 2^6 + b_5 \times 2^5 + b_4 \times 2^4 + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0$$

$$= (b_7 \times 2^1 + b_6) \times 2^6 + (b_5 \times 2^1 + b_4) \times 2^4 + (b_3 \times 2^1 + b_2) \times 2^2 + (b_1 \times 2^1 + b_0) \times 2^0$$

$$= a_3 \times 4^3 + a_2 \times 4^2 + a_1 \times 4^1 + a_0 \times 4^0$$

Nota che  $a_3, a_2, a_1, a_0$  sono compresi fra 0 e 3

## ➤ Algoritmo

- Raggruppa i bit 2 a 2 da destra
- Ad ogni gruppo fai corrispondere la cifra in base 4 (da 0 a 3)



# Esercizio 1: Soluzione

## ➤ Algoritmo

- Raggruppa i bit 2 a 2 da destra
- Ad ogni gruppo fai corrispondere la cifra in base 4  
(da 0 a 3)

- Quindi, partendo da  $(01000110)_2$  abbiamo
  - 01 00 01 10  $\rightarrow 1012_4$

$$0\ 0 = 0$$

$$0\ 1 = 1$$

$$1\ 0 = 2$$

$$1\ 1 = 3$$





# Quiz 2

Domanda:

Il valore di  $(10001011100)_2$  in ottale è

- $(4230)_8$
- $(2134)_8$
- $(1234)_8$
- Nessuno dei precedenti



# Quiz 2: Soluzione

Domanda:

Il valore di  $(10001011100)_2$  in ottale è

- $(4230)_8$
- $(2134)_8$
- $(1234)_8$
- Nessuno dei precedenti

Risposta:

Raggruppando in gruppi di 3 bit: 10 001 011 100  
Quindi il valore giusto è  $(2134)_8$

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



# Quiz 3

Domanda:

Il valore di  $(1011)_5$  in decimale è

- $(131)_{10}$
- $(521)_{10}$
- $(11)_{10}$
- Nessuno dei precedenti



# Quiz 3: Soluzione

Domanda:

Il valore di  $(1011)_5$  in decimale è

- $(131)_{10}$
- $(521)_{10}$
- $(11)_{10}$
- Nessuno dei precedenti

Risposta:  $5^3 + 5^1 + 5^0 = 125 + 5 + 1 = 131_{10}$

Il valore corretto è  $(131)_{10}$



# Esercizio 4

## Domanda:

Determinare quanti bit sono necessari per rappresentare l'insieme dei caratteri che comprende le lettere maiuscole e minuscole dell'alfabeto inglese, i numeri da 0 a 9 e i caratteri . + -



# Esercizio 4: Soluzione

## Domanda:

Determinare quanti bit sono necessari per rappresentare l'insieme dei caratteri che comprende le lettere maiuscole e minuscole dell'alfabeto inglese, i numeri da 0 a 9 e i caratteri . + -

## Risposta:

Poiché si tratta di  $26 \times 2 + 10 + 3 = 65$  simboli, sono necessari 7 bit

- Infatti con 7 bit posso rappresentare  $2^7 = 128$  oggetti distinti
- Invece con 6 bit posso rappresentare  $2^6 = 64$  oggetti distinti



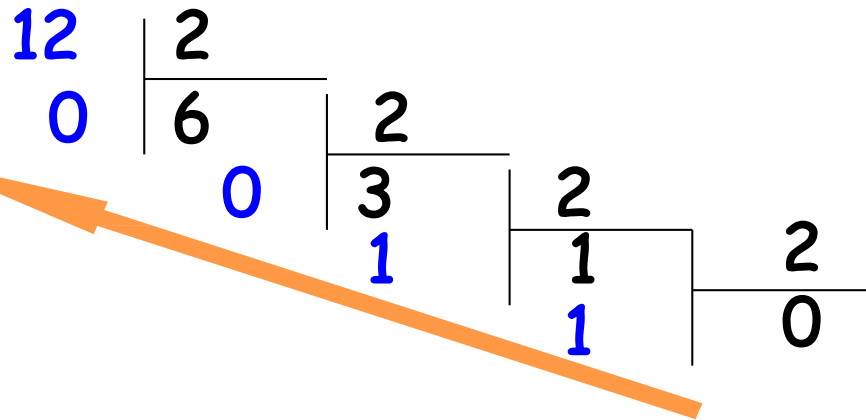
# Esercizio 5

➤ Convertire in binario il numero  $12,125_{10}$



# Esercizio 5: Soluzione

- Convertire in binario il numero  $12,125_{10}$ 
  - Convertiamo separatamente la parte intera e quella frazionaria



$$12_{10} = 1100_2$$

$$F=0,125$$

$$2 \times 0,125 = 0 + 0,25$$

$$2 \times 0,25 = 0 + 0,50$$

$$2 \times 0,50 = 1 + 0,00$$

$$0,125_{10} = 0,001_2$$

$$12,125_{10} = 1100,001_2$$





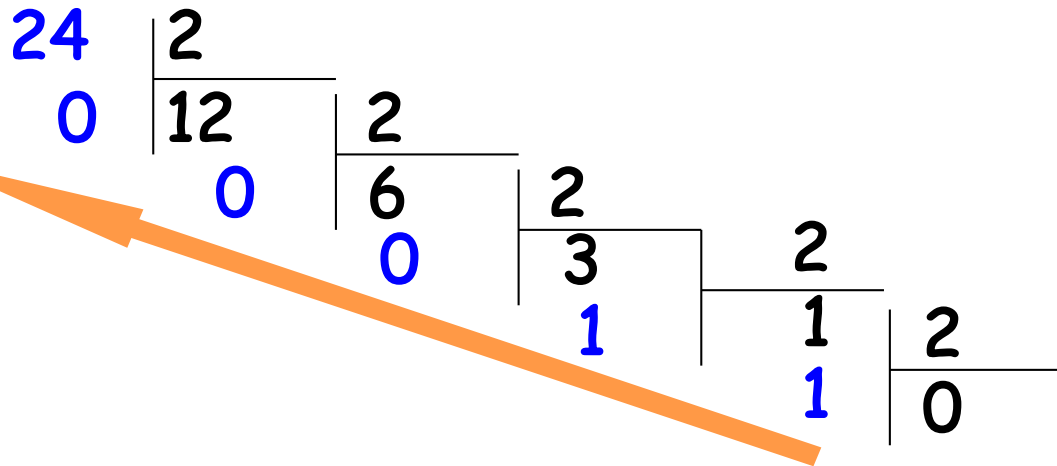
# Esercizio 6

➤ Convertire in binario il numero  $24,250_{10}$



# Esercizio 6: Soluzione

- Convertire in binario il numero  $24,250_{10}$ 
  - Convertiamo separatamente la **parte intera** e quella **frazionaria**



$$24_{10} = 11000_2$$

$$F = 0,250$$

$$\begin{aligned} 2 \times 0,250 &= 0 + 0,50 \\ 2 \times 0,50 &= 1 + 0,00 \end{aligned}$$

$$0,250_{10} = 0,01_2$$

$$24,250_{10} = 11000,01_2$$



# Esercizio 7

➤ Convertire in binario il numero  $0,07_{10}$



# Esercizio 7: Soluzione

➤ Convertire in binario il numero  $0,07_{10}$

$$F = 0,07_{10}$$

$$2 \times 0,07 = 0 + 0,14$$

$$2 \times 0,14 = 0 + 0,28$$

$$2 \times 0,28 = 0 + 0,56$$

$$2 \times 0,56 = 1 + 0,12$$

$$2 \times 0,12 = 0 + 0,24$$

$$2 \times 0,24 = 0 + 0,48$$

$$2 \times 0,48 = 0 + 0,96$$

$$2 \times 0,96 = 1 + 0,92$$

$$2 \times 0,92 = 1 + 0,84$$

$$2 \times 0,84 = 1 + 0,68$$

$$2 \times 0,68 = 1 + 0,36$$

$$2 \times 0,36 = 0 + 0,72$$

$$2 \times 0,72 = 1 + 0,44$$

.....

$$0,07_{10} = (0,000100011101)_2$$



# Esercizio 8

➤ Convertire in binario il numero  $0,06875_{10}$



# Esercizio 8: Soluzione

➤ Convertire in binario il numero  $0,06875_{10}$

$$F = 0,06875_{10}$$

$$2 \times 0,06875 = 0 + 0,1375$$

$$2 \times 0,1375 = 0 + 0,275$$

$$2 \times 0,275 = 0 + 0,55$$

$$2 \times 0,55 = 1 + 0,1$$

$$2 \times 0,1 = 0 + 0,2$$

$$2 \times 0,2 = 0 + 0,4$$

$$2 \times 0,4 = 0 + 0,8$$

$$2 \times 0,8 = 1 + 0,6$$

$$2 \times 0,6 = 1 + 0,2$$

$$2 \times 0,2 = 0 + 0,4$$

$$2 \times 0,4 = 0 + 0,8$$

$$2 \times 0,8 = 1 + 0,6$$

$$2 \times 0,6 = 1 + 0,2$$

.....

$$0,06875_{10} = (0,0001000110011)_2$$



# Esercizio 9

➤ Convertire in decimale il numero  $(0,0011101)_2$



# Esercizio 9: Soluzione

➤ Convertire in decimale il numero  $(0,0011101)_2$

$s=7$

$$F_{-6}=1/2=0,5$$

$$F_{-5}=(0+0,5)/2=0,25$$

$$F_{-4}=(1+0,25)/2=1,25/2=0,625$$

$$F_{-3}=(1+0,625)/2=1,625/2=0,8125$$

$$F_{-2}=(1+0,8125)/2=1,8125/2=0,90625$$

$$F_{-1}=(0+0,90625)/2=0,453125$$

$$F_0=(0+0,453125)/2=0,2265625$$

$$(0,0011101)_2=0,2265625_{10}$$





# Esercizio 10

➤ Convertire in decimale il numero  $(1101,00111)_2$



# Esercizio 10: Soluzione

- Convertire in decimale il numero  $(1101,00111)_2$ 
  - Convertiamo separatamente la **parte intera** e quella **frazionaria**

$$1101_2 \quad n=4$$

$$(0,00111)_2 \quad s=5$$

$$S_3 = a_3 = 1$$

$$S_2 = a_2 + 2S_3 = 1 + 2 = 3$$

$$S_1 = a_1 + 2S_2 = 0 + 6 = 6$$

$$S_0 = a_0 + 2S_1 = 1 + 12 = 13$$

$$1101_2 = 13_{10}$$

$$F_{-4} = 1/2 = 0,5$$

$$F_{-3} = (1 + 0,5)/2 = 1,5/2 = 0,75$$

$$F_{-2} = (1 + 0,75)/2 = 1,75/2 = 0,875$$

$$F_{-1} = (0 + 0,875)/2 = 0,4375$$

$$F_0 = (0 + 0,4375)/2 = 0,21875$$

$$(0,00111)_2 = 0,21875_{10}$$

$$(1101,00111)_2 = 13,21875_{10}$$



# Esercizio 11

➤ Convertire in decimale il numero  $(110,01)_2$



# Esercizio 11: Soluzione

- Convertire in decimale il numero  $(110,01)_2$ 
  - Convertiamo separatamente la parte intera e quella frazionaria

$$110_2 \quad n=3$$

$$(0,01)_2 \quad s=2$$

$$S_2 = a_2 = 1$$

$$S_1 = a_1 + 2S_2 = 1 + 2 = 3$$

$$S_0 = a_0 + 2S_1 = 0 + 6 = 6$$

$$F_{-1} = 1/2 = 0,5$$

$$F_0 = (0 + 0,5)/2 = 0,25$$

$$110_2 = 6_{10}$$

$$(0,01)_2 = 0,25_{10}$$

$$(110,01)_2 = 6,25_{10}$$



# Esercizio 12

- Trovare le basi  $b$  (interi positive) per cui valgono le seguenti uguaglianze
  - $21_b + 131_b = 24_b + 120_b + 4_b$
  - $(23_b)^2 - 240_b = (14_b)^2 + 60_b$



# Esercizio 12: Soluzione

- Trovare la base  $b$  (intera positiva) per cui vale la seguente uguaglianza

- $21_b + 131_b = 24_b + 120_b + 4_b$

- Poiché si tratta di un sistema posizionale pesato, si ha

$$2b+1+b^2+3b+1 = 2b+4+b^2+2b+4 \quad \text{da cui}$$

$$5b+2=4b+8 \quad \text{da cui}$$

$$b=6$$



# Esercizio 12: Soluzione

- Trovare la base  $b$  (intera positiva) per cui vale la seguente uguaglianza

- $(23_b)^2 - 240_b = (14_b)^2 + 60_b$

- Poiché si tratta di un sistema posizionale pesato, si ha

$$(2b+3)^2 - 2b^2 - 4b = (b+4)^2 + 6b \quad \text{da cui}$$

$$4b^2 + 12b + 9 - 2b^2 - 4b = b^2 + 8b + 16 + 6b \quad \text{da cui}$$

$$b^2 - 6b - 7 = 0 \quad \text{che ha soluzioni } b = -1 \text{ e } b = 7$$

- Essendo  $b$  un intero positivo, allora  $b=7$



# Esercizio 13

- Eseguire le seguenti operazioni in binario
  - $10011101 + 10101110$
  - $10011 + 1101$





# Esercizio 13: Soluzione

- Eseguiare la seguente operazione in binario
  - $10011101 + 10101110$

$$\begin{array}{r} 1 \quad 1111 \\ 10011101 + \\ 10101110 = \\ \hline 101001011 \end{array}$$



# Esercizio 13: Soluzione

- Eseguire la seguente operazione in binario
  - $10011 + 1101$

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1 \\ 10011 + \\ 1101 = \\ \hline 100000 \end{array}$$



# Esercizio 14

1. In accordo alla codifica ASCII Estesa (8 bit) scrivere in decimale e in binario le parole «Cosa» e «Papà»
2. In accordo alla codifica ASCII Estesa decodificare la stringa:  
01010100 01110101 01110100 01101111 01110010



# Codifica ASCII estesa

Primo gruppo  
(codici da 0 a 127)

Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char	Byte	Cod.	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	`
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	C	01100011	99	c
00000100	4	End of transmit	00100100	36	\$	01000100	68	D	01100100	100	d
00000101	5	Enquiry	00100101	37	%	01000101	69	E	01100101	101	e
00000110	6	Acknowledge	00100110	38	&	01000110	70	F	01100110	102	f
00000111	7	Audible bell	00100111	39	'	01000111	71	G	01100111	103	g
00001000	8	Backspace	00101000	40	(	01001000	72	H	01101000	104	h
00001001	9	Horizontal tab	00101001	41	)	01001001	73	I	01101001	105	i
00001010	10	Line feed	00101010	42	*	01001010	74	J	01101010	106	j
00001011	11	Vertical tab	00101011	43	+	01001011	75	K	01101011	107	k
00001100	12	Form Feed	00101100	44	,	01001100	76	L	01101100	108	l
00001101	13	Carriage return	00101101	45	-	01001101	77	M	01101101	109	m
00001110	14	Shift out	00101110	46	.	01001110	78	N	01101110	110	n
00001111	15	Shift in	00101111	47	/	01001111	79	O	01101111	111	o
00010000	16	Data link escape	00110000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00110001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00110010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00110011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00110100	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00110101	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00110110	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00110111	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00111000	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00111001	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00111010	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00111011	59	;	01011011	91	[	01111011	123	{
00011100	28	File separator	00111100	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111101	61	=	01011101	93	]	01111101	125	}
00011110	30	Record Separator	00111110	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111111	63	?	01011111	95	_	01111111	127	Del



# Codifica ASCII estesa

Secondo gruppo  
(codici da 128 a 255)

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



# Esercizio 14: Soluzione

1. In accordo alla codifica ASCII Estesa scrivere in decimale e in binario:

- Cosa :
  - 67 111 115 97
  - 01000011 01101111 01110011 01100001
- Papà:
  - 80 97 112 133
  - 01010000 01100001 01110000 10000101

2. 84 117 116 111 114, Tutor

