



Università degli Studi di Milano «La Statale»
Dipartimento di Informatica

La rappresentazione di testi: caratteri e stringhe

(parte A)

1



Rappresentiamo lettere, parole, testi: caratteri e stringhe

- Ora sappiamo come codificare numeri (naturali, interi, razionali)
- Anche le informazioni «alfabetiche» (o «testuali») vanno codificate!
- Idea base:
facciamo corrispondere ogni **carattere** (lettere) ad un **numero**
 - ▶ (in modo arbitrario... purché *standard*)
(cioè deciso una volta per tutte!)
 - ▶ per es: 1 $\iff$ A, 2 $\iff$ B
- una parola, o un testo, è solo una sequenza di caratteri (una «**stringa**»)
 - ▶ memorizzata come sequenza di numeri (interi senza segno!)
 - ▶ (compreso un carattere «spazio» per staccare le parole e un carattere «accapo» per staccare le righe)
- Problemi apparentemente banali:
 - ▶ quali caratteri? quanti caratteri?
 - ▶ quale corrispondenza caratteri-numeri?

Architettura degli elaboratori

- 2 -

Aritmetica binaria

2



La codifica dei caratteri

- Per il momento ci concentriamo sui caratteri base degli alfabeti occidentali, che, per motivi storici, sono stati i primi a essere codificati.
- Per gli usi più comuni, i caratteri usati sono relativamente pochi
 - ▶ 10 cifre numeriche 0,1,2,3,4,5,6,7,8,9
 - ▶ 26 minuscole: a,b,c, ...
 - ▶ 26 maiuscole: A,B,C, ...
 - ▶ 6 simboli di interpunzione (. , ? ! ; :)
 - ▶ pochissimi operatori matematici di uso comune (+ - < >)
 - ▶ apostrofi, accenti, virgolette (' " `)
 - ▶ un'altra manciata di simboli uso comune (° @ # \$ % & ' \ _ *)
 - ▶ In totale: meno di 100 caratteri:
7 bit (128 combinazioni) ci bastano per rappresentarli tutti.
- Nota: per i numeri esiste una codifica "naturale" (ad es. lo zero è rappresentato da un insieme di bit nulli) per i caratteri possiamo usare ad esempio l'ordine alfabetico.

Architettura degli elaboratori

- 4 -

Aritmetica binaria

4



La codifica dei caratteri: ASCII

- Con 7 bit, rappresento 128 caratteri diversi
- Basta fissare una corrispondenza fissa tra numeri appartenenti all'intervallo 0-127 e i caratteri.
- La cosa importante è che questa corrispondenza sia nota ed accettata da tutti (pena la scorretta interpretazione dei testi).
 - ▶ la codifica dei caratteri deve essere stabilita da uno standard.
- Lo standard più adottato è lo standard ASCII (persino prima dei computer! telescriventi, telegrafi...)

ASCII: American Standard Code for Information Interchange
(si pronuncia «àschii», ma in Ita a volte anche «àscii»)

Architettura degli elaboratori

- 5 -

Aritmetica binaria

5

La tabella ACII

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

6

 La tabella ASCII

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	NUL				EOT			BEL	BS	TAB	LF			CR		
1-																
2-	space	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Architettura degli elaboratori - 7 - Aritmetica binaria

7



I caratteri ASCII «di controllo» cioè «non stampabili»

- Caratteri non visibili ma con un effetto sul testo, come:
- Sono una trentina (codici da 0x00 a 0x1F, e 0xFF)
- Originalmente, controllavano le funzionalità di dispositivi come telescriventi & telegrafi
- Molti di questi sono oggi senza senso per la rappresentazione dei testi, e sono inutilizzati.
 - ▶ es:
 - BEL** «suona un campanello» :-D
 - DEL** e **BS** «cancella l'ultimo carattere» (in due modi diversi) e molti altri
 - ▶ purtroppo ce li portiamo dietro, solo per motivi storici ☹
 - ▶ consumano spazio di rappresentazione (cioè lasciano meno codici liberi per rappresentare i caratteri utili)
- Alcuni hanno ancora degli usi, come «accapo» (

Architettura degli elaboratori

- 8 -

Aritmetica binaria

8



I caratteri ASCII «di controllo» ancora usati (nella rappresentazione di testi)

- **TAB**: tabulazione (spazio orizzontale, per allineare);
- I due caratteri che segnalano la fine di una riga di testo:
 - ▶ **CR** – carriage return (codice 13):
originalmente: «riporta la testina scrivente all'inizio della riga»
 - ▶ **LF** – line feed (codice 10):
originalmente: «scorri il foglio della pagina giù di una riga»
 - ▶ Oggi (purtroppo) per «andare accapo»...
 - alcuni standard (Linux, MacOS) usano il 2°
 - altri (Windows) usano ancora entrambi i caratteri (in sequenza)
 - (altri ancora, in disuso (es. Commodore) usavano solo il 1°)
- **NUL**, il carattere nullo (codice 0):
 - ▶ spesso usato ad es per segnalare la fine di una stringa in memoria
 - ▶ stringa = una sequenza di caratteri in memoria
- **EOT** – end of text (fine del testo)
 - ▶ a volte usato per segnalare la fine di un file di testo;

9



ASCII – una nota

- Per decenni, è stato l'unico standard diffuso per rappresentare testi.
- L'informatica ne porta ancora (e ne porterà sempre) le tracce. Infatti:
 - ▶ perché per la mail si usa il simbolo @?
 - ▶ perché per gli hashtag si usa #?
 - ▶ perché nei path dei file, o negli URL dei siti web, si usano / e \?
 - ▶ perché in HTML nei tag si aprono con < e chiudono con >?
 - ▶ perché in molti URL appare la «tilde» ~ (per indicare homepage)?
 - ▶ perché per dire «citazione», es nei forum o nelle mail, si usa > blah?
 - ▶ perché il prompt dei comandi è denotata da >, ad es: C:\>?
 - ▶ perché il «pipe command» di Unix ha quella sintassi, con |?
 - ▶ perché si sono così diffusi gli *smiley*, come :-)?
quando (oggi) ci sono gli *emoticons*, come ☺?
 - ▶ perché il *dash* – si trova scritto spesso come un meno -?
 - ▶ perché spesso si trova scritto *perche* ' al posto di *perché*?
 - ▶ perché il celebre sito slashdot si chiama proprio slashdot?

Architettura degli elaboratori

- 10 -

Aritmetica binaria

10



ASCII – una nota

- Per decenni, è stato l'unico standard diffuso per rappresentare testi.
- L'informatica ne porta ancora (e ne porterà sempre) le tracce.
- Perché, in quasi tutti i linguaggi di programmazione (dal C al Go)...
 - ▶ si usano parentesi (), [], {}, <> (con la più varia semantica)?
 - ▶ si usano spesso i simboli \$, & e % (con la più varia semantica)?
 - ▶ la divisione si denota con / (e non, mettiamo, con ÷)?
 - ▶ la moltiplicazione si denota con * (e non, mettiamo, con × o •)?
 - ▶ il minore-o-uguale si denota con <= (e non con ≤)?
 - ▶ *diverso-da* si denota con != oppure <> (e non con ≠)?
 - ▶ le virgolette sono aperte (e chiuse!) con " o ' (e mai con, p.e., « e »)?
 - ▶ per gli operatori logici si usa per es & | ! (invece che, per es, ∧ ∨ ¬)?
 - ▶ le frecce si scrivono con sequenze: -> o => (e non → o ⇒)?
 - ▶ per la radice quadrata si usano lettere: *sqrt*, e non √?
 - ▶ i quantificatori si scrivono ad es. come *forEach* e non ∀?
 - ▶ il valore di +infinito si scrive come *+inf* e non +∞
 - ▶ alla x alla seconda si scrive *x^2* e non x²?
 - ▶ Etc.

Architettura degli elaboratori

- 11 -

Aritmetica binaria

11



Il codice ASCII non ha :_(

- Lettere accentate (o nessun altro modificatore di lettera)
- Simboli matematici, eccetto i pochissimi basilari (+, -, <, >)
- Lettere greche
 - ▶ neanche quelle di uso comune, come pi-greco o epsilon
- Anzi... nessun alfabeto che non sia quello latino di base!
 - ▶ no: cirillico, giapponese, coreano, etc
- Simboli per valute che non siano il dollaro
- Interpunzioni rovesciate (per es. usate in spagna)
- Emoji ☺
- etc...

Insomma la codifica ASCII è (a malapena) sufficiente solo per

- i testi più basilari
- testi espressi in linguaggi formali
(e non per caso: sono linguaggi progettati attorno a queste limitazioni)

12



Esercizi di riepilogo

In ASCII:

- Quale sequenza di 7 bit rappresenta la lettera «erre minuscola»?
- Quale sequenza di 7 bit rappresenta la lettera «erre maiuscola»?
- Quale sequenza di 7 bit rappresenta la lettera «effe minuscola»?
- Quale sequenza di 7 bit rappresenta la lettera «effe maiuscola»?
- Noti un pattern, nelle sequenze di 7 bit che rappresentano le lettere minuscole e maiuscole?
(Fai altri esempi se necessario, e verifica di avere ragione)
- Scrivi in esadecimale il BYTE che rappresenta il punto esclamativo
(il MSB – cioè il Most Significant Bit, che non viene utilizzato, vale 0)

13