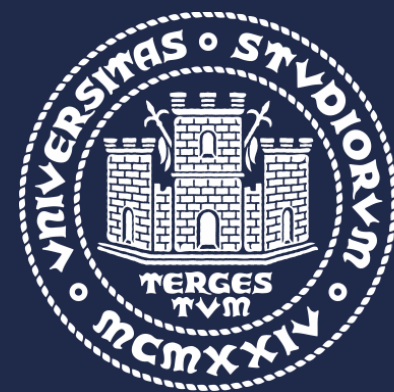




UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Rappresentazione dell'informazione (4)

Standard IEEE 754 e ASCII/Unicode

Prof.ssa Giulia Cisotto

giulia.cisotto@units.it

Trieste, 25 febbraio 2026

STANDARD IEEE 754

IEEE Computer Society (Institute of Electrical and Electronics Engineers) definisce lo “IEEE standard for binary floating arithmetic” noto anche come **IEEE 754** nel 1985.

Specifica il **formato**, le **operazioni**, le **conversioni** tra i diversi formati floating point e quelle tra i diversi sistemi di numerazione, il trattamento delle **eccezioni**

Nel 1989 IEEE 754 diventa uno standard diventa uno **standard internazionale**.

Formato non proprietario, ossia **non dipendente dall'architettura** del calcolatore

VIRGOLA MOBILE



La massa dell'elettrone è
0.0000000000000000000000000000000091 Kg



La massa della terra è
597360000000000000000000000 kg

Requisiti ideali:

- Convenzione per rappresentazione univoca per numeri così diversi
- Costo limitato di rappresentazione nel sistema binario (significa memoria!)

VIRGOLA MOBILE



La massa dell'elettrone è
0.0000000000000000000000000000000091 Kg



La massa della terra è
59736000000000000000000000 kg

Passiamo alla notazione scientifica

$$9.1 \cdot 10^{-31}$$

$$5.9736 \cdot 10^{24}$$

VIRGOLA MOBILE

Floating point

Standard IEEE 754

segno

esponente

$$\mathcal{N} = (-1)^S \cdot 1.M \cdot B^{\pm E}$$

MANTISSA

BASE

NOTA. Per poter rappresentare sia esponenti positivi che negativi, si usa un **bias** (per evitare di dover memorizzare un altro bit di segno).

+/ -	e	e	...	e	m	m	...	m
segno	esponente				mantissa			

Più bit qui per range maggiore

Più bit qui per maggior precisione

VIRGOLA MOBILE: PRECISIONE

Floating point

Standard IEEE 754



32 bit in tutto → **SINGOLA (o semplice) PRECISIONE**



64 bit in tutto → **DOPPIA PRECISIONE**

VIRGOLA MOBILE: FORME NORMALIZZATE

Floating point

$$\mathcal{N} = \pm M \cdot B^{\pm E} = \boxed{363,4 \cdot 10^{34}} = \boxed{36,34 \cdot 10^{35}} = \boxed{3,634 \cdot 10^{36}}$$



forme equivalenti

forma *NORMALIZZATA*
(1 cifra prima della virgola)

Forme *NORMALIZZATE*

$$\mathcal{N} = \pm X, YYY \cdot 10^{\pm E}$$

Base 10

$$\mathcal{N} = \textcircled{1} YY \dots Y_2 \cdot 2^{\pm EE \dots E_2}$$

Base 2

Bit sempre a 1 (bit implicito)

Massimizza la precisione

RAPPRESENTARE UN NUMERO BINARIO IN FORMATO IEEE 754

Algoritmo:

Il numero binario di partenza è scritto come **001XX.X**

1. **Determinare il bit di segno** (0 per numeri positivi, 1 per numeri negativi).



Dobbiamo sapere com'è rappresentato il numero (virgola fissa signed/unsigned o CA2)

Se il numero è «con segno», allora il MSB è il bit di segno.

2. **Spostare la virgola** fino a ottenere un numero della forma **1xxx** × 2ⁿ.

3. **Scrivere la mantissa in binario**, rimuovendo il primo 1 implicito.

4. **Calcolare l'esponente da rappresentare: $n + bias$** . Il bias sarà 127 (per singola precisione) oppure 1023 (per doppia precisione). Convertire in formato binario il risultato.

5. **Scrivere la rappresentazione finale in formato IEEE 754.**

+/-	e	e	...	e	m	m	...	m
segno	esponente				mantissa			

RAPPRESENTARE UN NUMERO BINARIO IN FORMATO IEEE 754

Esempio:

Supponiamo che il numero in formato IEEE 754 abbia le seguenti parti:

- **Segno** $S=1$
- **Mantissa** $M=101011$ (sono 6 bit)
- **Base** $B=2$
- **Esponente (reale)** $E=0101$ (sono 4 bit)

! Ricordarsi il **bias (+127)** nella codifica dell'esponente)

A che numero binario in forma normalizzata corrisponde?

Segno	Esponente (codifica)	Mantissa
1	10000100	101011000000000000000000



-



5_{10}



101011

$$N = -1.101011 \cdot 2^5$$

RAPPRESENTARE UN NUMERO *DECIMALE* IN FORMATO IEEE 754

Esempio: rappresentare il numero **-10.375** in formato IEEE 754 in *singola precisione*

Il numero è negativo, quindi il bit di segno è 1: **Segno S = 1**

Convertire 10 in binario: $10_{10} = 1010_2$

Convertire 0.375 in binario con la moltiplicazione per 2:

$0.375 \times 2 = 0.75 \rightarrow$ parte intera = 0

$$0.375_{10} = 0.011_2$$

$0.75 \times 2 = 1.5 \rightarrow$ parte intera = 1

$0.5 \times 2 = 1.0 \rightarrow$ parte intera = 1 (**fine conversione**)

Il numero iniziale è convertito in formato binario: $(-)10.375_{10} = (-)1010.011_2$

Passare alla notazione scientifica binaria: $1010.011_2 = 1.010011_2 \times 2^3$

L'esponente reale è 3. Aggiungere il bias (singola precisione): **E = 3 + 127 = 130**

Convertire 130 in binario: $130_{10} = 10000010_2$

Estrarre la **mantissa** dal numero in formato normalizzato $1.010011_2 \times 2^3$: **M = 010011₂**

1 10000010 010011000000000000000000

APPLICAZIONI DELLO STANDARD IEEE 754

- **Hardware:** IEEE 754 è implementato in quasi tutti i processori moderni per gestire operazioni in virgola mobile con elevata precisione.

- **CPU (Central Processing Unit)** → Intel, AMD, ARM implementano l'unità FPU (Floating Point Unit).
- **GPU (Graphics Processing Unit)** → NVIDIA, AMD e Intel utilizzano IEEE 754 per calcoli grafici e scientifici.
- **DSP (Digital Signal Processor)** → Utilizzato in processori per audio, immagini e telecomunicazioni.

- **Linguaggi di programmazione (Python, C, Java, ...)** nella codifica di *float* e *double*. Gli sviluppatori possono scrivere codice che funziona in modo identico su diverse piattaforme.

- **AI e Deep Learning:** IEEE 754 fornisce una precisione sufficiente per il deep learning riducendo il consumo di memoria.

- **TensorFlow, PyTorch** → Le reti neurali usano float32 (IEEE 754 a 32-bit).
- **AI in GPU** → NVIDIA utilizza fp16 (IEEE 754 a 16-bit) per accelerare i calcoli.
- **Big Data Analytics** → Elaborazione numerica su milioni di dati.

- Sistemi embedded e IoT (smartphone, robotica, sensori).

- Database e sistemi informativi (SQL, Big Data Analytics).

ESERCIZIO 1

Quale numero in singola precisione rappresentano i seguenti 32 bit?

1 10000001 010000000000000000000000

6 bit 21 bit

- Segno negativo (-)
- Esponente $e = 2^7 + 2^0 - 127 = 129 - 127 = 2$
- Mantissa $m = (1+) 2^{-2} = 1.25$

Quindi il numero rappresentato è $-1.25 \times 2^2 = -5$

ESERCIZIO 2

Qual è la rappresentazione a singola precisione del numero 8.5

- Segno positivo (0)
- 8.5 in binario è $1000.1 \cdot 2^0 = 1.0001 \cdot 2^3$
- Esponente e: $3 + 127 = 130 = 10000010$
- Mantissa m: 000100000000000000000000

0 10000010 000100000000000000000000

Virgola fissa vs virgola mobile

Scegliere correttamente che tipo di rappresentazione utilizzare è fondamentale per **ottimizzare la precisione e/o l'uso della memoria nei sistemi embedded, DSP e applicazioni numeriche.**



ERRORI DI APPROSSIMAZIONE

Rappresentando un **numero reale** n in virgola mobile si commette un errore di approssimazione.

In realtà viene rappresentato un numero razionale n' con un numero limitato di cifre significative:

ERRORE ASSOLUTO: $e_A = n - n'$

ERRORE RELATIVO: $e_R = e_A / n = (n - n') / n$

L'ordine di grandezza dell'**errore assoluto** dipende dal numero di cifre significative e dall'ordine di grandezza del numero.

L'ordine di grandezza dell'**errore relativo** dipende solo dal numero di cifre significative.

Rappresentazione di caratteri di testo

Possiamo associare a ogni **carattere** (quale lettera minuscola, lettera maiuscola, vocale accentata e segno di interpunzione) un numero.

I caratteri possono essere rappresentati in **3 diversi standard**:

- **ASCII standard**: 1 carattere è rappresentato con 7 bit per un totale di 128 simboli rappresentabili (quali cifre, lettere maiuscole e lettere minuscole);
- **ASCII estesa**: 1 carattere è rappresentato con 8 bit rappresentabili fino a 256 simboli (i caratteri in più sono usati per esempio per caratteri accentati);
- **UNICODE**: 1 carattere è rappresentato con un numero maggiore di bit (tra 8 e 32 bit per carattere).

ASCII standard

ASCII standard contiene:

- 26 + 26 **lettere** (maiuscole + minuscole)
- 10 **cifre decimali** (da 0 a 9)
- segni di **interpunzione**
- **caratteri di controllo**
- Le cifre sono ordinate per valore
- Le lettere maiuscole sono ordinate alfabeticamente
- Le lettere minuscole sono ordinate alfabeticamente (e sono a distanza fissa dalle maiuscole)

[Link alla tabella ASCII](#)

codifiche					codifiche					codifiche					codifiche				
Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	`
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	\$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
25	0x19	031	0011001	EM	57	0x39	071	0111001	9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
26	0x1A	032	0011010	SUB	58	0x3A	072	0111010	:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
27	0x1B	033	0011011	ESC	59	0x3B	073	0111011	;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
28	0x1C	034	0011100	FS	60	0x3C	074	0111100	<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	
29	0x1D	035	0011101	GS	61	0x3D	075	0111101	=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
30	0x1E	036	0011110	RS	62	0x3E	076	0111110	>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
31	0x1F	037	0011111	US	63	0x3F	077	0111111	?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

ASCII standard

- Dal 0 a 31 sono dei caratteri di controllo per periferiche
- Da 32 a 47 vari caratteri
- da 48 a 57 cifre decimali
- Da 58 a 64 vari caratteri
- Da 65 a 90 lettere maiuscole dell'alfabeto
- Da 91 a 96 vari caratteri
- Da 97 a 122 lettere minuscole dell'alfabeto
- Da 123 a 127 vari caratteri

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	`
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	\$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
25	0x19	031	0011001	EM	57	0x39	071	0111001	9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
26	0x1A	032	0011010	SUB	58	0x3A	072	0111010	:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
27	0x1B	033	0011011	ESC	59	0x3B	073	0111011	;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
28	0x1C	034	0011100	FS	60	0x3C	074	0111100	<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	
29	0x1D	035	0011101	GS	61	0x3D	075	0111101	=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
30	0x1E	036	0011110	RS	62	0x3E	076	0111110	>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
31	0x1F	037	0011111	US	63	0x3F	077	0111111	?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

ASCII esteso

Con 1 byte=8 bit ($2^8=256$) è possibile realizzare 256 diverse combinazioni.

Vengono definiti tutti i caratteri dell'ASCII standard con le stesse identiche codifiche, ma uno 0 davanti per l'8^a bit.

ASCII control characters			ASCII printable characters			Extended ASCII characters										
00	NULL	(Null character)	32	space	64	@	96	·	128	Ç	160	à	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ó
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ù	195	ł	227	Õ
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ö
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	å	165	ñ	197	†	229	Ö
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	ä	166	*	198	ā	230	μ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	*	199	Ä	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	Ł	232	þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	©	201	ł	233	ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	™	202	Ł	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	%	203	ł	235	Ü
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	%	204	Ł	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	ı	205	—	237	Ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ā	174	«	206	Ł	238	—
15	SI	(Shift In)	47	/	79	O	111	o	143	Ā	175	»	207	□	239	·
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	▯	208	ø	240	•
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▯	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▯	210	É	242	—
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ó	179	▯	211	Ê	243	¼
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ô	180	▯	212	Ë	244	½
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	õ	181	▯	213	İ	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	▯	214	ı	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ü	183	▯	215	İ	247	•
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	▯	216	ı	248	•
25	EM	(End of medium)	57	9	89	Y	121	y	153	Œ	185	▯	217	ı	249	—
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ů	186	▯	218	ı	250	•
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	▯	219	ı	251	•
28	FS	(File separator)	60	<	92	\	124		156	£	188	▯	220	ı	252	•
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	▯	221	ı	253	•
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	ı	254	■
31	US	(Unit separator)	63	?	95	_			159	f	191	γ	223	ı	255	nbsp
127	DEL	(Delete)														

Parole

Le parole sono sequenze di caratteri.

La codifica binaria della parola
«informatica» in ASCII esteso è:

01101001 i	01101110 n	01100110 f	01101111 o	01110010 r
01101101 m	01100001 a	01110100 t	01101001 i	01100011 c
01100001 a				

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	.
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	\$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
									9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
									:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
									;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
									<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	
									=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
									>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
									?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

ASCII esteso

La tabella ASCII estesa varia in base alla zona geografica di utilizzo e al software utilizzato.

Le principali estensioni previste dall'ISO 8859 sono:

- **ISO-8859-1 (Latin-1)**, utilizzato nella Zona Europea Occidentale
- **ISO-8859-2 (Latin-2)**, utilizzato zona Europea Orientale (Serbia, Albania, Ungheria, Romania)
- **ISO-8859-3 (Latin-3)**, utilizzato nell' Europea del Sud (Malta), include l'Esperanto
- **ISO-8859-4 (Latin-4)**, obsoleto
- **ISO-8859-5 (Part 5, Cyrillic)**, alfabeto Cirillico
- **ISO-8859-6 (Part 6, Arabic)**, alfabeto Arabo
- **ISO-8859-7 (Part 7, Greek)**, alfabeto Greco
- **ISO-8859-8 (Part 8, Hebrew)**, alfabeto Ebraico

ASCII vs UNICODE

ASCII è un codice accettato da tutti i computer.

Usato dai tempi delle telescriventi durante la prima guerra mondiale.

Tuttavia ***non considera i caratteri internazionali di numerose lingue straniere.***

Per ovviare a tale problematica, è stata introdotta un'ulteriore codifica, ossia

UNICODE

UNICODE

Unicode è uno standard per la rappresentazione di caratteri ed è un'evoluzione dello standard ASCII.

- Codifica tutti i caratteri utilizzati nelle principali lingue del mondo
- *Indipendente dalla lingua, dal sistema operativo e dal programma utilizzato*
- Inizialmente rappresentato con codifica a 16 bit, poi esteso a 24 e 32 bit
- Disporre di 32 bit significa avere 4 miliardi di caratteri diversi codificabili!
- Unicode è in continua evoluzione e continua ad aggiungere sempre più caratteri

UNICODE

Un carattere UNICODE è caratterizzato dal suo codice numerico, detto code point, solitamente rappresentato con 8 cifre esadecimali

Esempi:

- «fi» è rappresentato dal codice 0x0000FB01 (*esadecimale!*)
- il simbolo “do doppio diesis strumentale” della notazione musicale greca antica è 0x0001D235

Con UNICODE, è possibile creare e gestire senza troppa pena documenti multilingue:

Α, Δ, Ё, ρ, ρ, あ, 叶, 葉

In particolare, tutti gli standard W3C (incluso HTML) supportano UNICODE

A marzo 2024 è stato presentato l'ultima versione UNICODE 16.0

IL PROBLEMA DELLA CODIFICA

UNICODE può codificare 4.294.967.296 caratteri distinti

- Ogni carattere occupa 32 bit (contro gli 8 delle altre codifiche); i documenti richiedono quindi 4 volte lo spazio
- La quasi totalità dei documenti usa da 60 a 1000 caratteri, per cui basterebbero da 6 a 10 bit.
- Per ovviare a questo problema e garantire maggiore compatibilità con S.O. e applicazioni che non sono in grado di gestire 32 bit per carattere, **UNICODE definisce vari formati di codifica più compatti**

CODIFICA UNICODE UTF-8

UTF-8 (8-bit UCS/Unicode Transformation Format) è una codifica a lunghezza variabile fra una sequenza di valori a 8 bit e una sequenza di caratteri UNICODE

- I primi 128 caratteri di UNICODE (0-7F), equivalenti ai caratteri ASCII, sono codificati con il loro codice “naturale”
- **Tutti gli altri caratteri sono codificati con due, tre o quattro valori a 8 bit (byte)**

CODIFICA UNICODE UTF-8: APPLICAZIONI

Nel linguaggio di programmazione Java (e derivati), le stringhe sono codificate con UTF-8; i programmi Java sono quindi in grado di gestire nativamente UNICODE.

- I file system Macintosh, DVD e alcuni su UNIX usano UTF-8 per i nomi dei file.
- Gli standard relativi al Web e alla e-mail richiedono che un programma compatibile supporti almeno UTF-8 come standard di codifica.
- I programmi che trattano testi ASCII sono generalmente UTF-8 compatibili.

CURIOSITÀ

Il sito principale relativo a UNICODE è <http://www.unicode.org>

- la pagina <https://symbl.cc/en/unicode/table/> è particolarmente affascinante
- Alcune emoji codificate in unicode a luglio 2023



Materiale per la lezione

- Appendici A e B Tanenbaum (numeri binari, virgola mobile, IEEE 754)
- Tabella ASCII

Prossima lezione: 26 febbraio, h.14:00, aula 4C