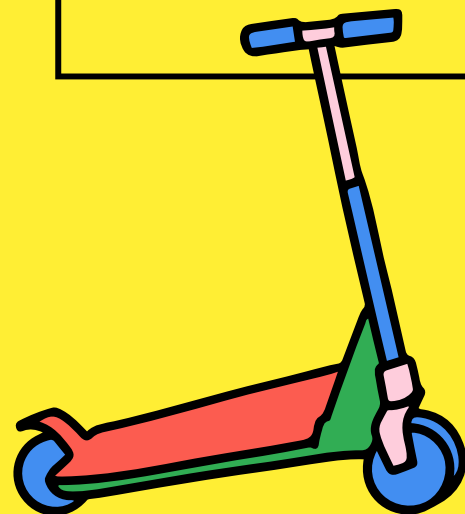
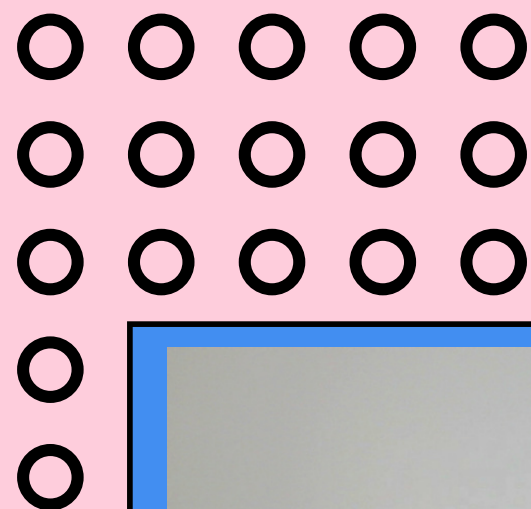


CRITTOGRAFIA

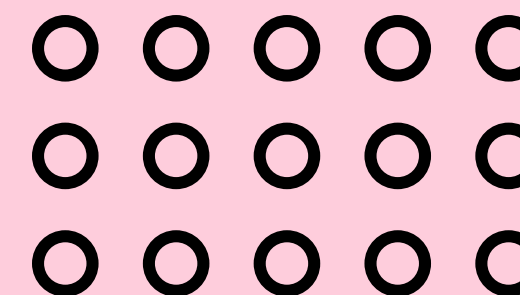
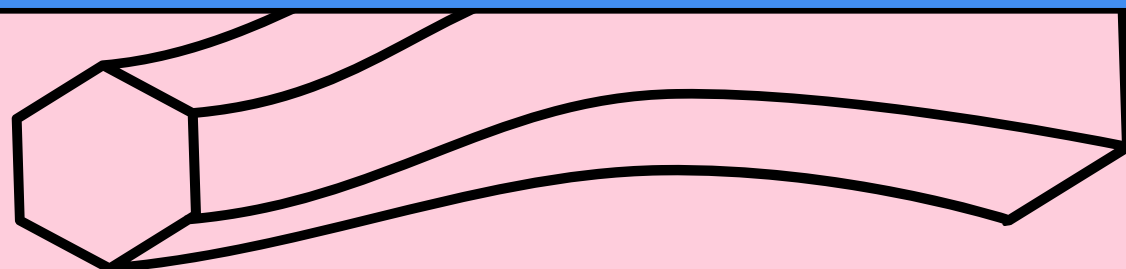
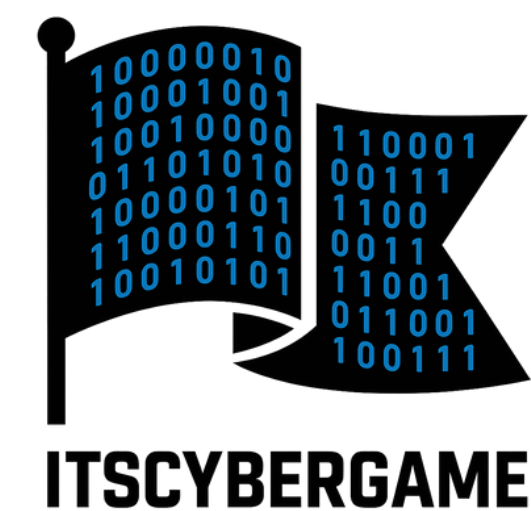
Di Piubelli Alberto





Chi sono io?

edulife
Fondazione ETS

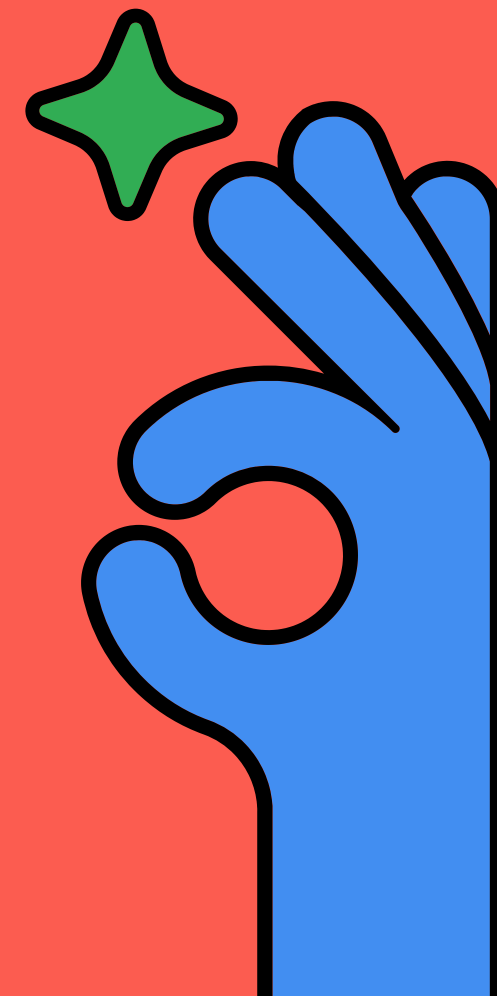
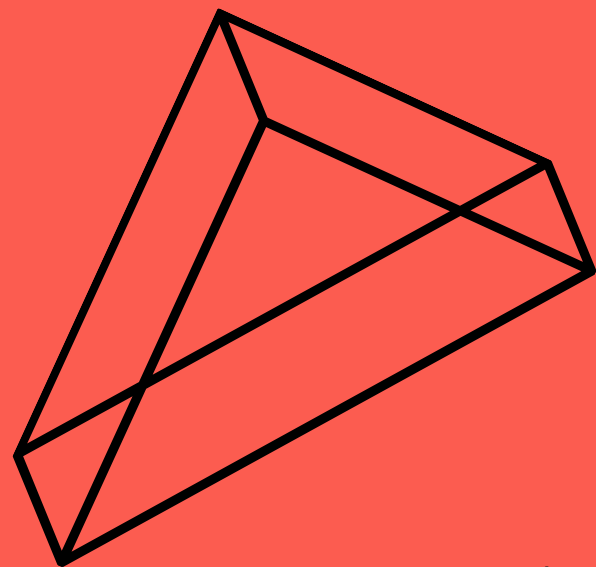


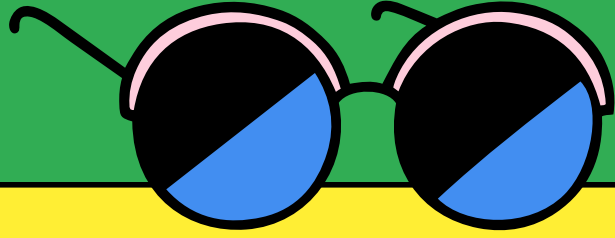
WELCOME TO CLASS!



Today's Agenda

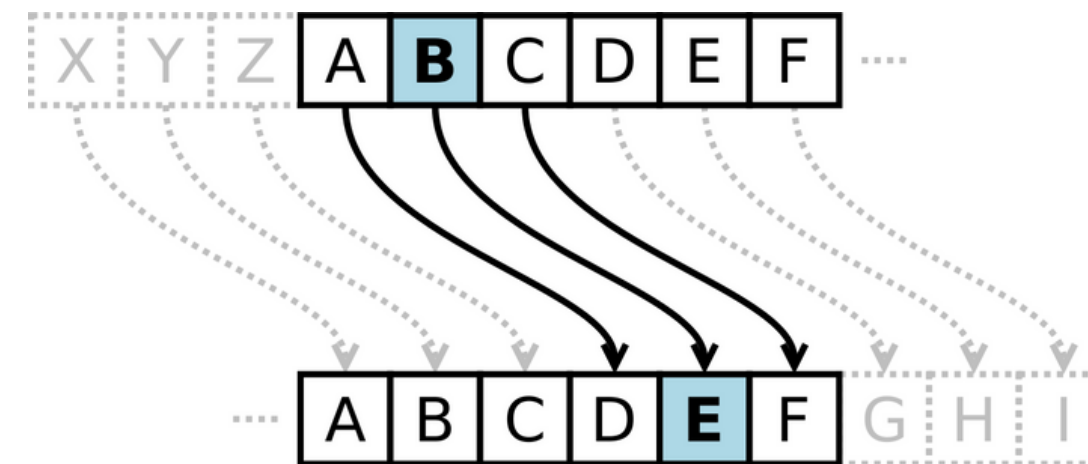
- Storia della crittografia
 - Encoding di base
- In quante basi sai contare?
 - ASCII e ASCII esteso
 - XOR
- Quiz finale

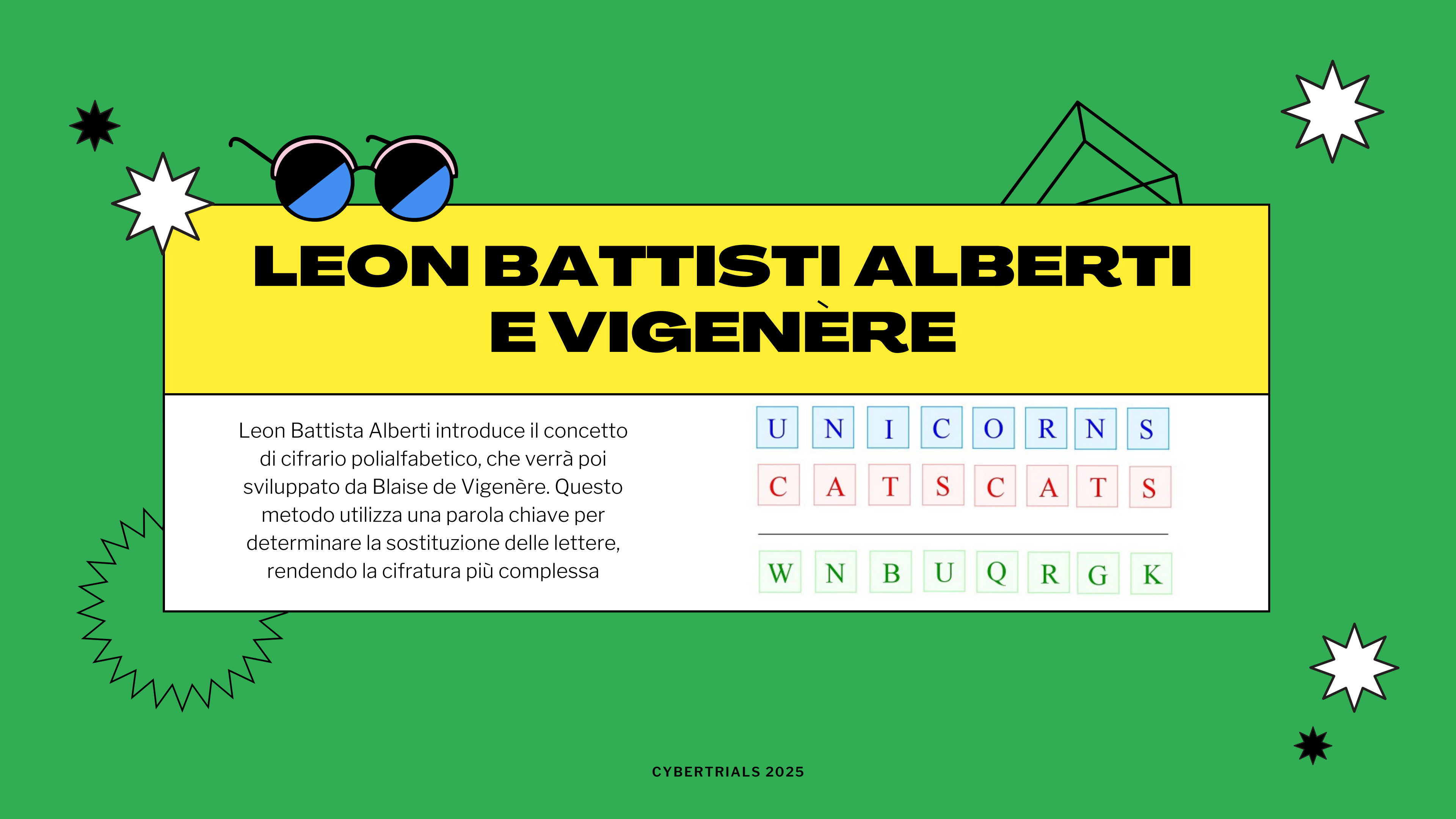




CIFRARIO DI CESARE

Il generale romano voleva inviare messaggi segreti ai suoi generali. Così inventò un metodo semplice: ogni lettera veniva spostata di 3 posti nell'alfabeto. "CIAO" diventava "FLDR". Questo è uno dei primi cifrari della storia!





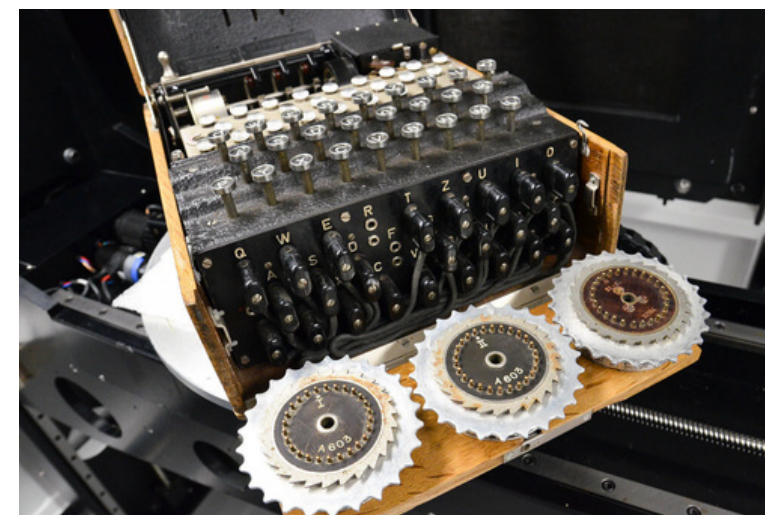
LEON BATTISTI ALBERTI E VIGENÈRE

Leon Battista Alberti introduce il concetto di cifrario polialfabetico, che verrà poi sviluppato da Blaise de Vigenère. Questo metodo utilizza una parola chiave per determinare la sostituzione delle lettere, rendendo la cifratura più complessa

U	N	I	C	O	R	N	S
C	A	T	S	C	A	T	S
W	N	B	U	Q	R	G	K

ENIGMA - BOMBA

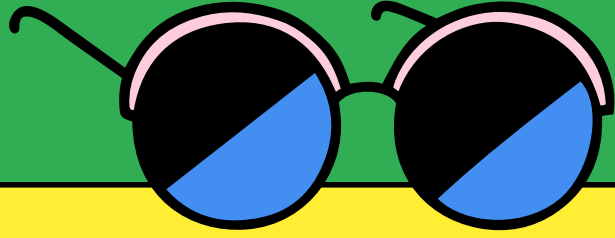
Durante la Seconda Guerra Mondiale, i tedeschi utilizzano la macchina Enigma per cifrare i messaggi militari. Alan Turing e il suo team a Bletchley Park riescono a decifrare i codici, contribuendo significativamente alla vittoria degli Alleati



CRITTOGRAFIA A CHIAVE PUBBLICA

Whitfield Diffie e Martin Hellman
introducono il concetto di crittografia a
chiave pubblica, permettendo lo scambio
sicuro di informazioni senza la necessità di
condividere una chiave segreta in anticipo





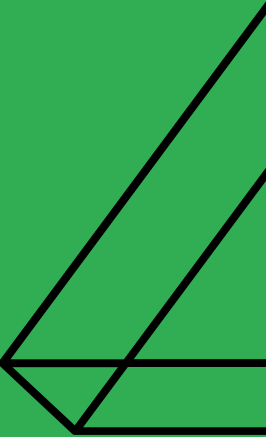
CRITTOGRAFIA END-TO-END

WhatsApp implementa la crittografia end-to-end, garantendo che solo il mittente e il destinatario possano leggere i messaggi scambiati, proteggendo la privacy degli utenti.





MA PRIMA DI TUTTO.. RISPETTARE LA C.I.A.



Confidenzialità (Confidentiality)

Assicura che solo le persone autorizzate possano accedere alle informazioni.
In crittografia: significa cifrare i dati in modo che chi non ha la chiave non possa leggere il contenuto.

Esempio: un messaggio su WhatsApp cifrato end-to-end non può essere letto da un hacker che lo intercetta.

Integrità (Integrity)

Garantisce che i dati non siano stati alterati durante il trasferimento o l'archiviazione.
In crittografia: si usa spesso una funzione di hash per verificare che il contenuto sia rimasto intatto.

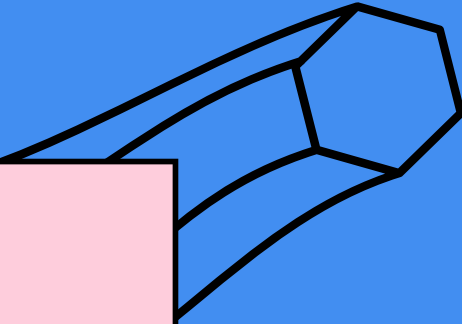
Esempio: se scarichi un file e l'hash corrisponde a quello originale, sai che non è stato modificato.

Disponibilità (Availability)

Autenticità: permette di verificare chi ha inviato il messaggio e che non sia stato falsificato.
In crittografia: si usano firme digitali e certificati.

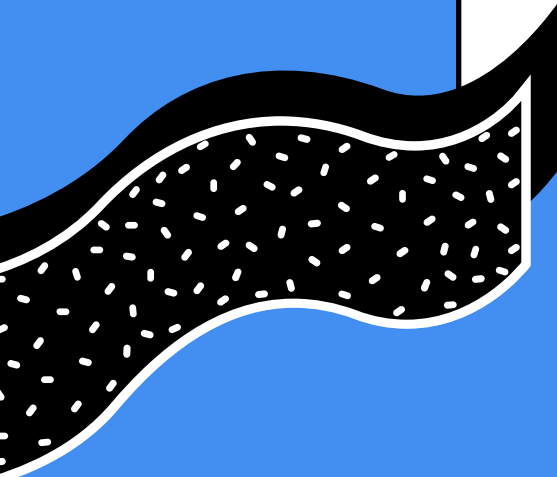
Disponibilità: (nell'ambito della sicurezza in generale) assicura che i dati e i servizi siano accessibili quando necessario.





ENCODING DI BASE

**102, 108, 97, 103, 123, 117, 103, 104, 95, 78, 117,
109, 66, 51, 114, 53, 95, 52, 49, 114, 51, 52, 100,
121, 125**





102

partiamo dal binario

Il codice binario è un sistema numerico che utilizza solo due simboli: 0 e 1. In informatica, rappresenta qualsiasi tipo di dato (testo, immagini, suoni) traducendolo in sequenze di bit comprensibili ai computer.

1	1	0	0	1	1	0
2⁶	2⁵	2⁴	2³	2²	2¹	2⁰

64+32+0+0+4+2+0=102

Tabella ASCII

La tabella ASCII (American Standard Code for Information Interchange) è uno standard che assegna a ogni carattere (lettere, numeri, simboli) un numero compreso tra 0 e 127.

Serve per permettere ai computer di rappresentare il testo: ad esempio, il carattere A corrisponde al numero 65, mentre a è 97.

Riprendiamo quindi il problema iniziale:

102, 108, 97, 103, 123, 117, 103, 104, 95, 78, 117, 109, 66, 51, 114, 53, 95, 52, 49, 114, 51, 52, 100, 121, 125

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

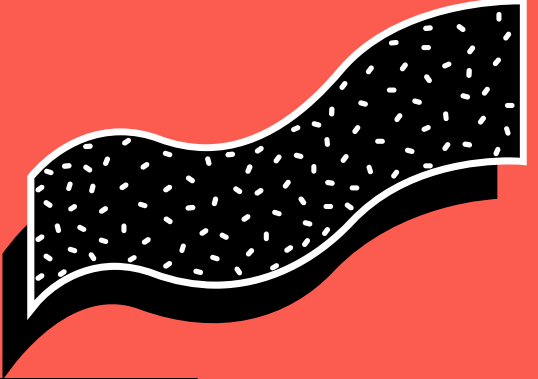
MA NON È TUTTO QUI

Espansione dell'ASCII: da 7 a 8 bit
 Con 8 bit si possono rappresentare 256 valori
 Questo ha permesso la nascita di:
 ASCII esteso (che aggiunge simboli speciali, lettere accentate, ecc.)
 Più tardi: UTF-8, che usa 8 bit (e oltre) per codificare i caratteri Unicode

Znaky č. 0 až 127 jsou shodné s kódem ASCII															
128	Ç	144	É	160	á	176	☐	192	ƒ	208	đ	224	Ó	240	
129	ü	145	Í	161	í	177	☐	193	⌞	209	Đ	225	ß	241	ˆ
130	é	146	Î	162	ó	178	☐	194	⌟	210	Ď	226	Ô	242	˜
131	â	147	Ï	163	ú	179		195	⌡	211	Ě	227	Ň	243	˘
132	ã	148	Ö	164	À	180	⌢	196	–	212	ď	228	ň	244	˙
133	û	149	Ł	165	ą	181	Á	197	⌣	213	Ň	229	ń	245	§
134	ć	150	ł	166	ż	182	Â	198	Ă	214	Í	230	Š	246	÷
135	ç	151	Ś	167	ž	183	Ě	199	ă	215	Î	231	š	247	˚
136	ť	152	ś	168	Ę	184	Ş	200	Ł	216	ě	232	Ř	248	°
137	ë	153	Ö	169	ę	185	⌥	201	⌦	217	Ј	233	Ú	249	ˆ
138	Õ	154	Ü	170	ˆ	186	⌧	202	⌨	218	Г	234	í	250	˙
139	õ	155	Ť	171	ž	187	〈	203	〉	219	☐	235	Ů	251	ũ
140	î	156	ť	172	Č	188	⌫	204	⌭	220	☐	236	ý	252	Ř
141	Ž	157	Ł	173	ş	189	Ž	205	=	221	Ţ	237	Ý	253	ř
142	Ä	158	×	174	«	190	ž	206	⌮	222	Ů	238	ţ	254	■
143	Ć	159	č	175	»	191	ƒ	207	¤	223	☐	239	˘	255	



**MI SORGE UNA
DOMANDA,
SAPETE TUTTE
CONTARE?**



SÌ?

IN QUANTE BASI?

Le diverse basi

Quando contiamo normalmente — 1, 2, 3, ... — stiamo usando la **base 10**, detta anche decimale.

Ma in informatica esistono anche altre basi fondamentali, come la **base 2** (binario) e la **base 16** (esadecimale). Vediamo cosa significa!

che cos'è una base

Una base è il numero di simboli diversi che usiamo per scrivere i numeri.

In base 10 usiamo le cifre da 0 a 9 → perché abbiamo 10 simboli

In base 2 (binario) usiamo solo 0 e 1 → 2 simboli

In base 16 (esadecimale) usiamo:

la base Esadecimale (HEX)

Decimale: 217

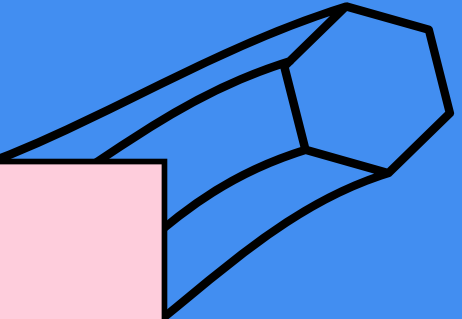
Binario: 11011001

Gruppi: 1101 1001

Decimali: 13 9

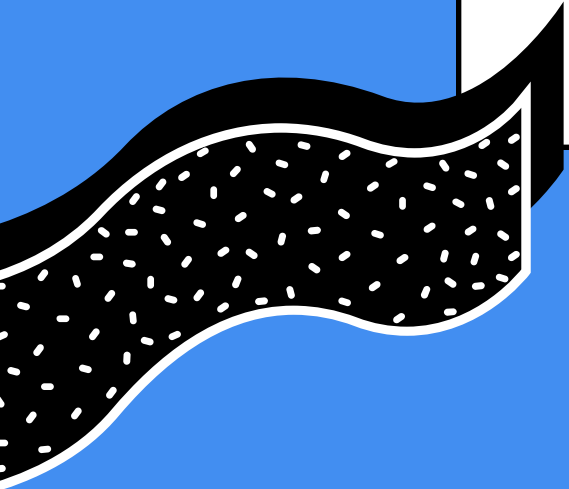
Hex: D 9 → Risultato: D9

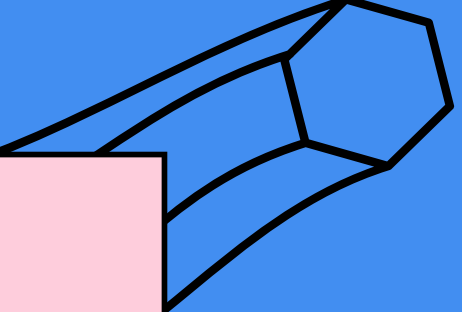
DECIMALE	Esadecimale	BINARIO
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111



ENCODING ESADECIMALE

**666c61677b68337834646563696d616c5f6334
6e5f62335f41424144424142457d**





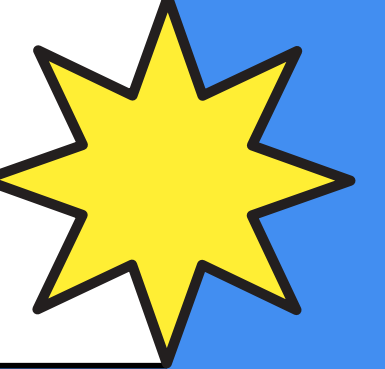
XOR

m1=

**158bbd7ca876c60530ee0e0bb2de20ef8af95b
c60bdf**

m2=

**73e7dc1bd30ef6576f883e79edaa48dcd58e6a
a82aa2**



Lo XOR

Pensa allo XOR come a un interruttore della luce:

Premilo una volta = luce ON

Premilo di nuovo = luce OFF

Lo XOR funziona così: applica due volte la stessa chiave = torna tutto come prima

Applicazione Pratica

Supponiamo:

Messaggio: 10101010 (A)

Chiave segreta: 11001100 (B)

$A \oplus B = C = 01100110$

$C \oplus B = A = 01100110 \oplus 11001100 = 10101010$

Torniamo al problema

m1 = 158bbd7ca876c60530ee0e0bb2de20ef8af95bc60bdf

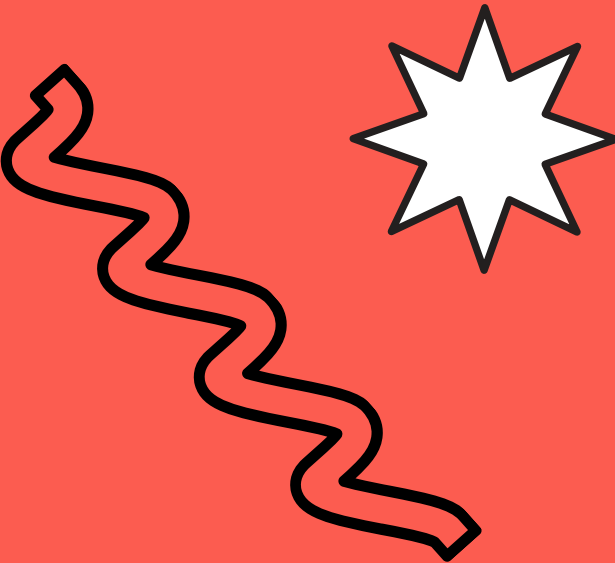
m2 = 73e7dc1bd30ef6576f883e79edaa48dcd58e6aa82aa2

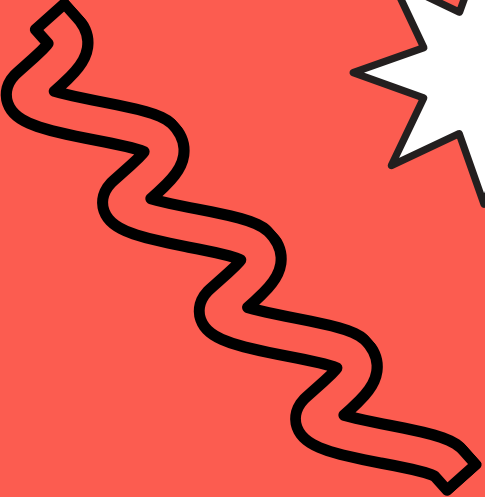


Symbol	A	B	Q
	0	0	0
	1	0	1
	0	1	1
	1	1	0
$Q = A \oplus B$			

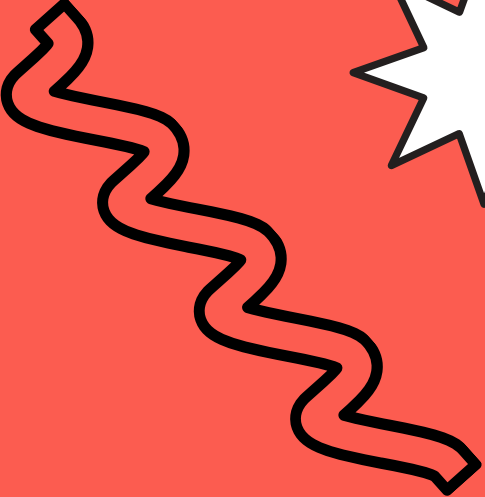


DOMANDE?





**ADESSO
GIOCHIAMO!!**



**GRAZIE PER
L'ATTENZIONE:)**