

Università degli Studi di Bologna

Area Medica

Insegnamento: INFORMATICA

Lezione: 01

Docente: [Ciro Polizzi - e-mail: ciro.polizzi@unibo.it](mailto:ciro.polizzi@unibo.it)

A.A 2021 - 2022

Computer Science an overview

- **Sistema di numerazione binaria - bit, byte e operatori logici**
- **Informatica di base: architettura di Von Neumann e macchina di Turing.**
- **Hardware di un calcolatore.**
- **Sistemi di codifica e rappresentazione dell'informazione.**
- **Scienza e Tecnologia dei supporti di memoria (analogici e digitali).**
- **Sistemi Multitasking, Ipermediale, Multimediale, Multimodale**
- **Applicazioni: Editor, Spreadsheet, DataBase...**
- **Sistemi operativi,**
- **Algoritmi e programmazione.**
- **Reti informatiche e protocolli.**
- **Strumenti digitali certificati: firma digitale, hash, marca temporale, PEC, PCT.**
- **Virus e antivirus.**

Sistema di numerazione binaria

(bit –byte kbyte ..)



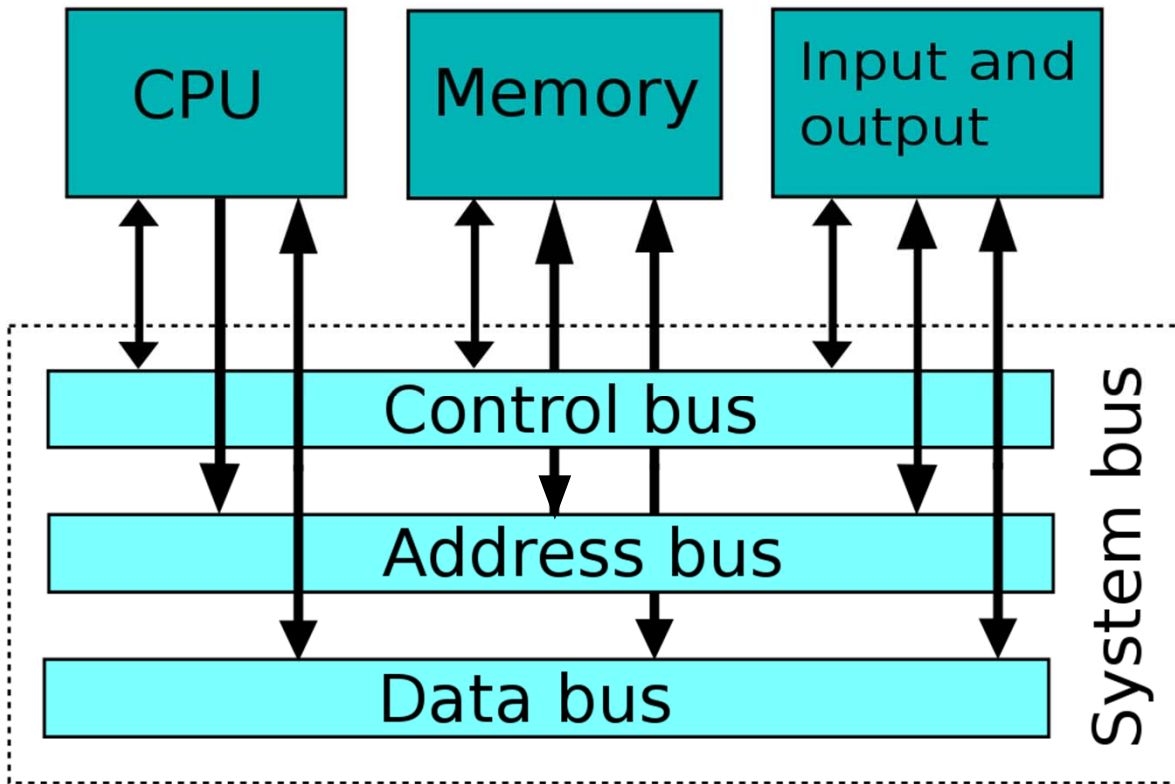
John von Neumann



John von Neumann (Budapest 1903 – Washington 1957)
matematico – fisico -informatico

È considerato uno dei più grandi matematici della storia moderna e una delle personalità scientifiche preminenti del XX secolo

Architettura di Von Neumann



- 1.CPU Unità centrale di elaborazione (unità ALU e unità di controllo)
- 2.Unità di memoria RAM
- 3.Unità di INPUT
- 4.BUS

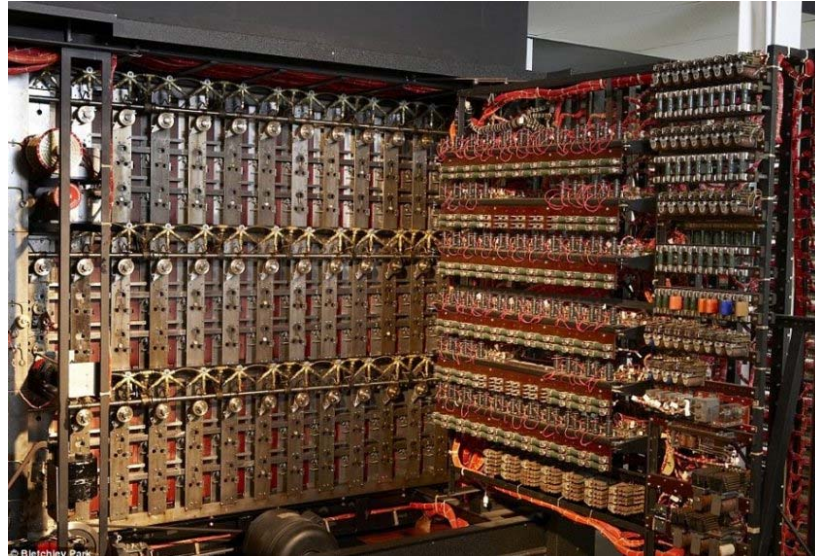
Alan Mathison Turing



(Londra nel 1912 Regno Unito 1954)

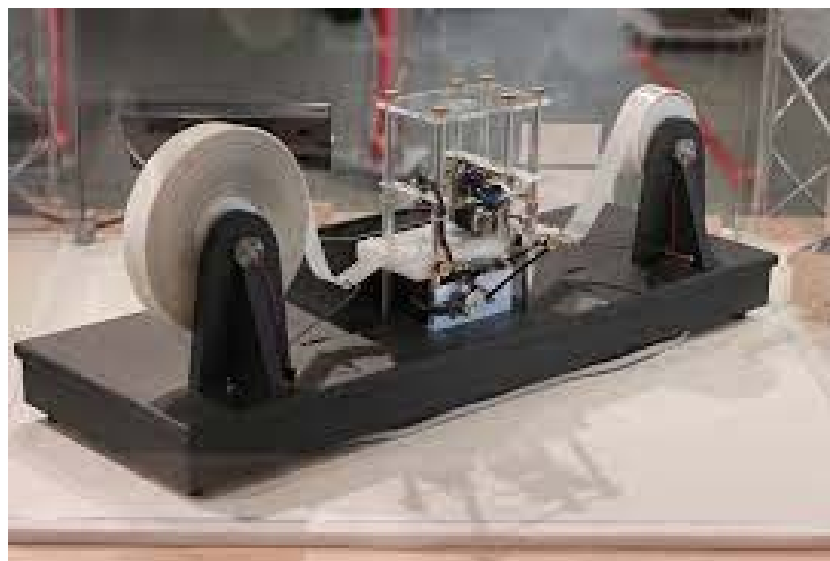
Alan Mathison Turing è stato un matematico, logico, crittografo e filosofo britannico, considerato uno dei padri dell'informatica e uno dei più grandi matematici del XX secolo

Macchina di Turing



Modello di agente di calcolo adatto a simulare la logica di qualsiasi algoritmo computazionale. Un sistema astratto che, opportunamente programmato, era capace di eseguire ogni tipo di operazione

Macchina di Turing

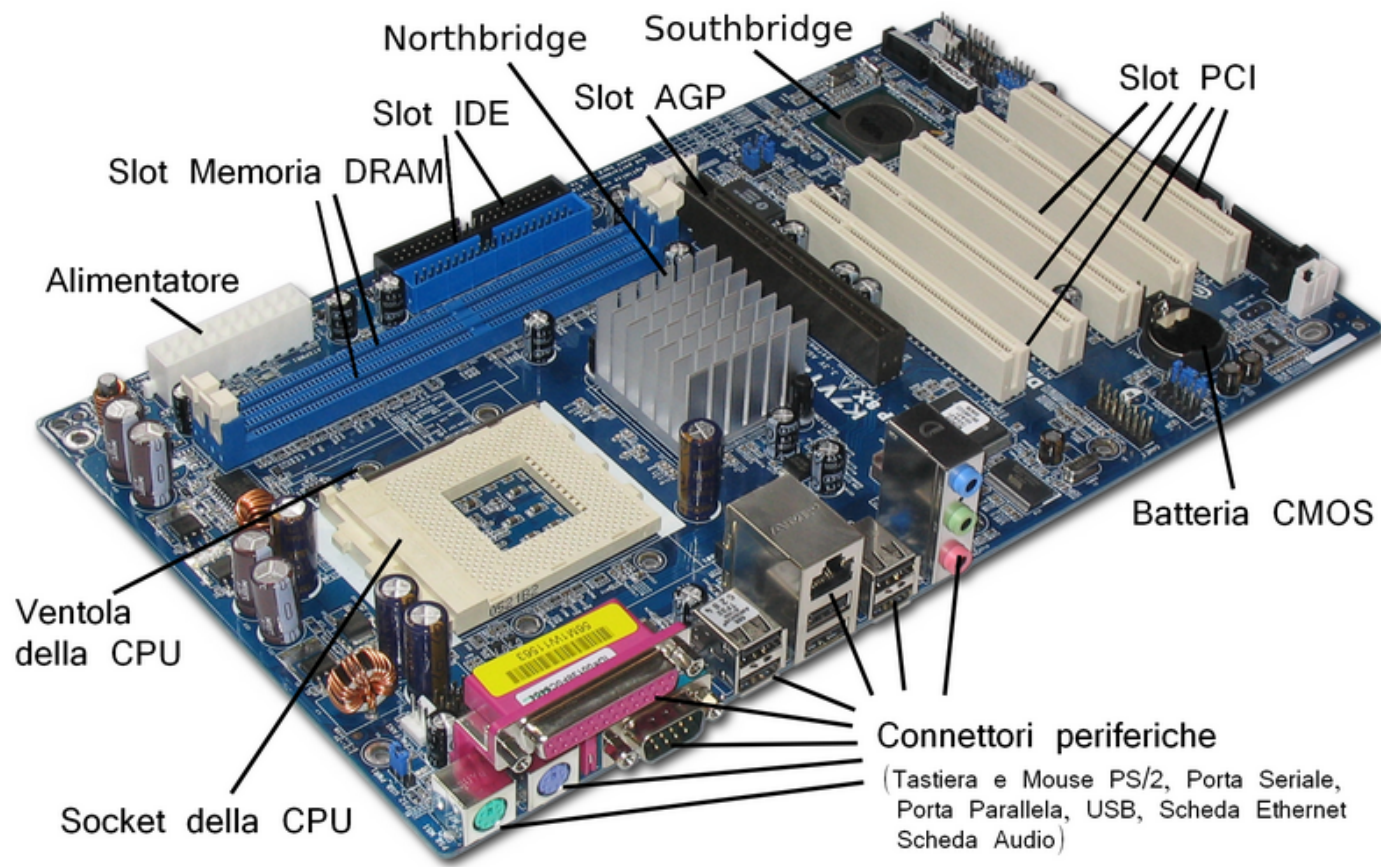


la macchina di Turing è formata da un'unità di controllo contenente un programma con un numero finito di istruzioni, da un nastro di lunghezza illimitata suddiviso in celle e da un'unità di lettura e scrittura sul nastro in grado di spostarsi avanti e indietro di un numero qualsiasi di celle e di leggere e scrivere in una qualsiasi delle celle un simbolo di un alfabeto prefissato.

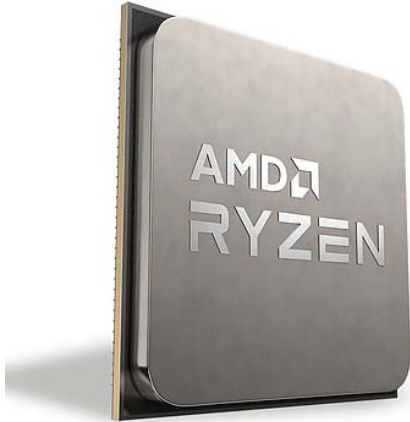
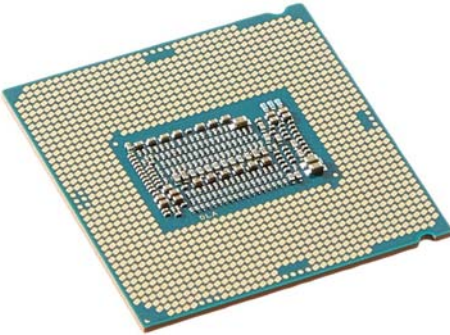
Hardware in un calcolatore



Scheda madre



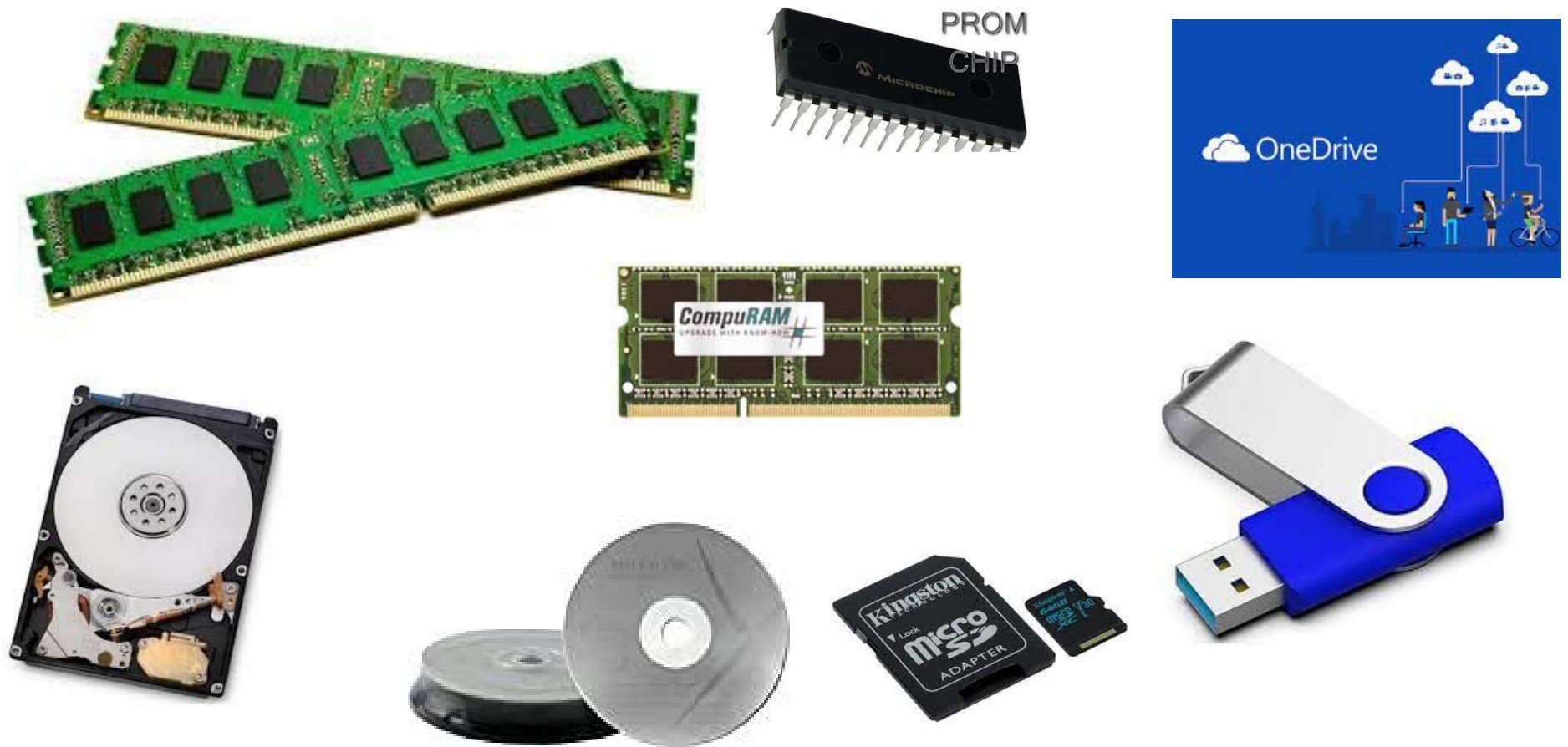
CPU



Periferiche



Memorie



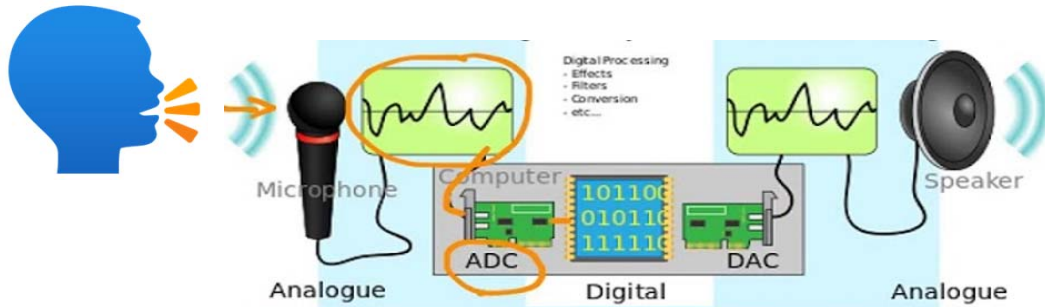
Dispositivi di INPUT OUTPUT



Rappresentazione dell'informazione – Sistemi di codifica A/D e D/A - (convergenza digitale)



→ 1 01001011011101...



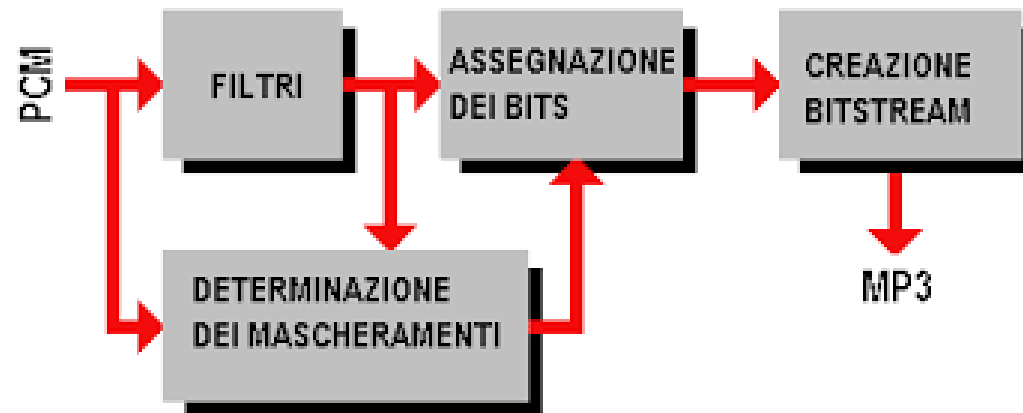
Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char	Byte	Cod	Char
00000000	0	Null	00100000	32	Spc	01000000	64	@	01100000	96	.
00000001	1	Start of heading	00100001	33	!	01000001	65	A	01100001	97	a
00000010	2	Start of text	00100010	34	"	01000010	66	B	01100010	98	b
00000011	3	End of text	00100011	35	#	01000011	67	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	00100000	48	0	01010000	80	P	01110000	112	p
00010001	17	Device control 1	00100001	49	1	01010001	81	Q	01110001	113	q
00010010	18	Device control 2	00100010	50	2	01010010	82	R	01110010	114	r
00010011	19	Device control 3	00100011	51	3	01010011	83	S	01110011	115	s
00010100	20	Device control 4	00101000	52	4	01010100	84	T	01110100	116	t
00010101	21	Neg. acknowledge	00101001	53	5	01010101	85	U	01110101	117	u
00010110	22	Synchronous idle	00101010	54	6	01010110	86	V	01110110	118	v
00010111	23	End trans. block	00101011	55	7	01010111	87	W	01110111	119	w
00011000	24	Cancel	00101100	56	8	01011000	88	X	01111000	120	x
00011001	25	End of medium	00101101	57	9	01011001	89	Y	01111001	121	y
00011010	26	Substitution	00101110	58	:	01011010	90	Z	01111010	122	z
00011011	27	Escape	00101111	59	;	01011011	91	[01111011	123	{
00011100	28	File separator	00111000	60	<	01011100	92	\	01111100	124	
00011101	29	Group separator	00111001	61	=	01011101	93]	01111101	125	}
00011110	30	Record Separator	00111010	62	>	01011110	94	^	01111110	126	~
00011111	31	Unit separator	00111011	63	?	01011111	95	_	01111111	127	Del



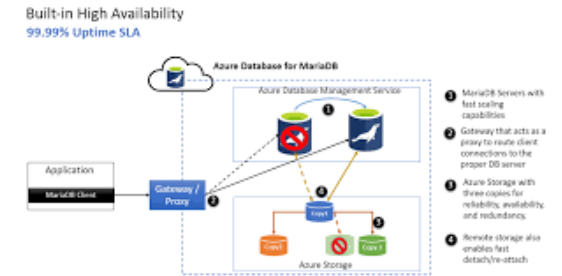
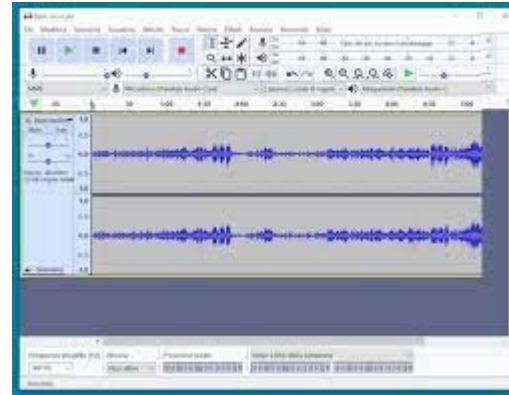
ADC → 1010010110111010010001... → DAC



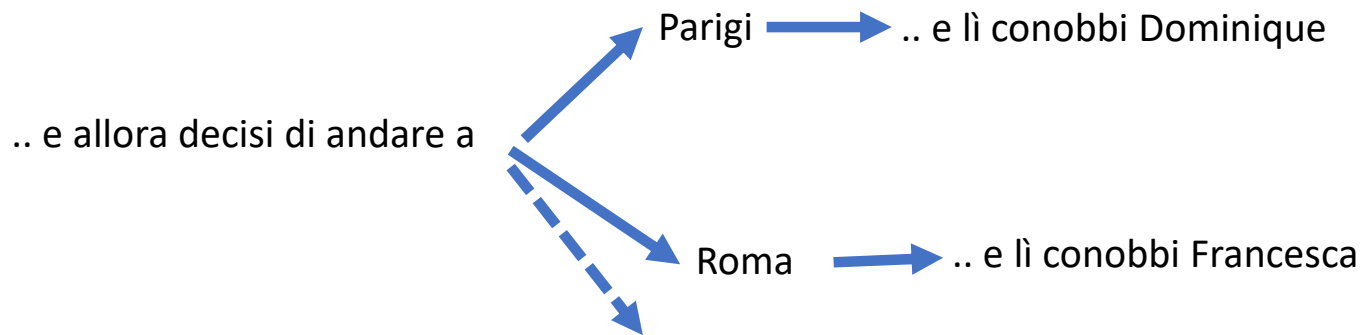
Sistemi di compressione loss e lossless



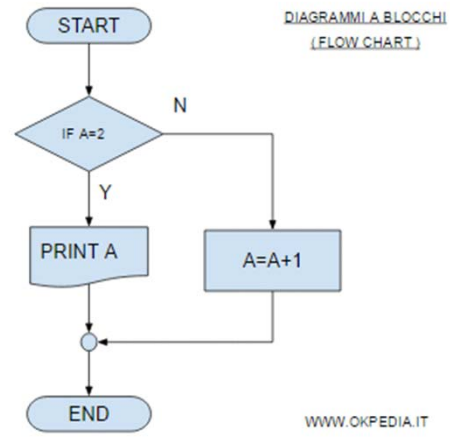
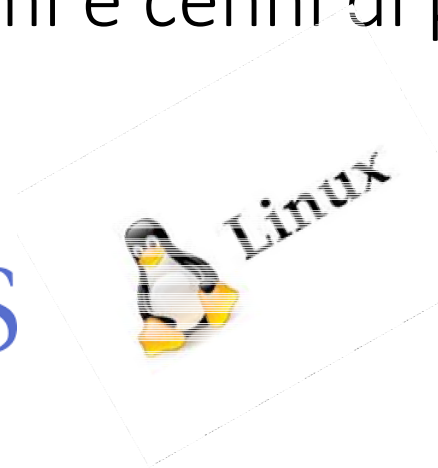
Editor, Spreadsheet, di presentazione, DataBase ecc.



Sistema Multitasking, Ipermediale, Multimediale, Multimodale.



Sistemi operativi, algoritmi e cenni di programmazione



```
m = int(input('Inserire m: '))
n = int(input('Inserire n: '))

c = s = 0

for i in range(1, min(m, n)+1):
    if m%i == 0 and n%i == 0:
        print(i)
        c+=1
        s+=i
```

Internet, reti informatiche, protocolli di trasmissione http, p2p e applicativi.

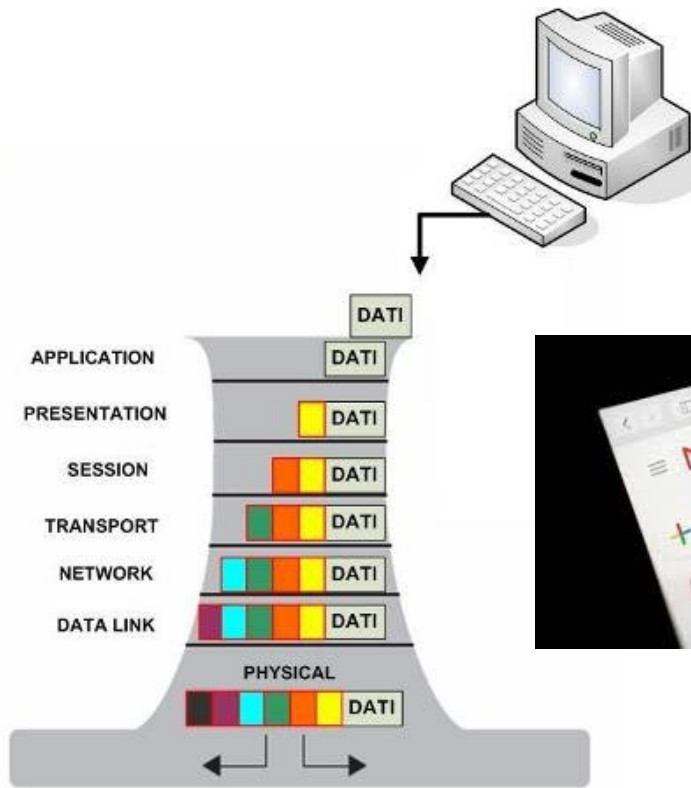
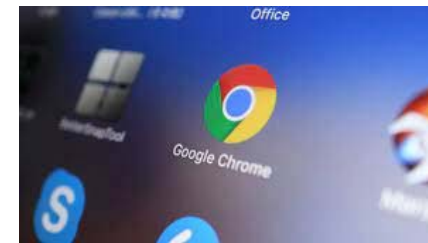


Figura 9



Firma digitale, hash (crittografia) , marca temporale, PEC, PCT.



**HASH
L'IMPRONTA
DIGITALE DI
UN FILE**

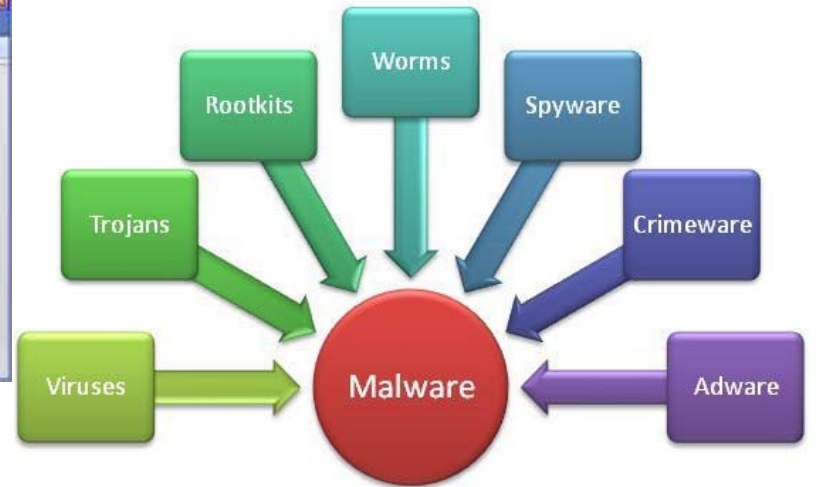


ARUBA PEC



**Marca
Temporale**

Virus e antivirus e firewall



Codifica e rappresentazione dell'informazione

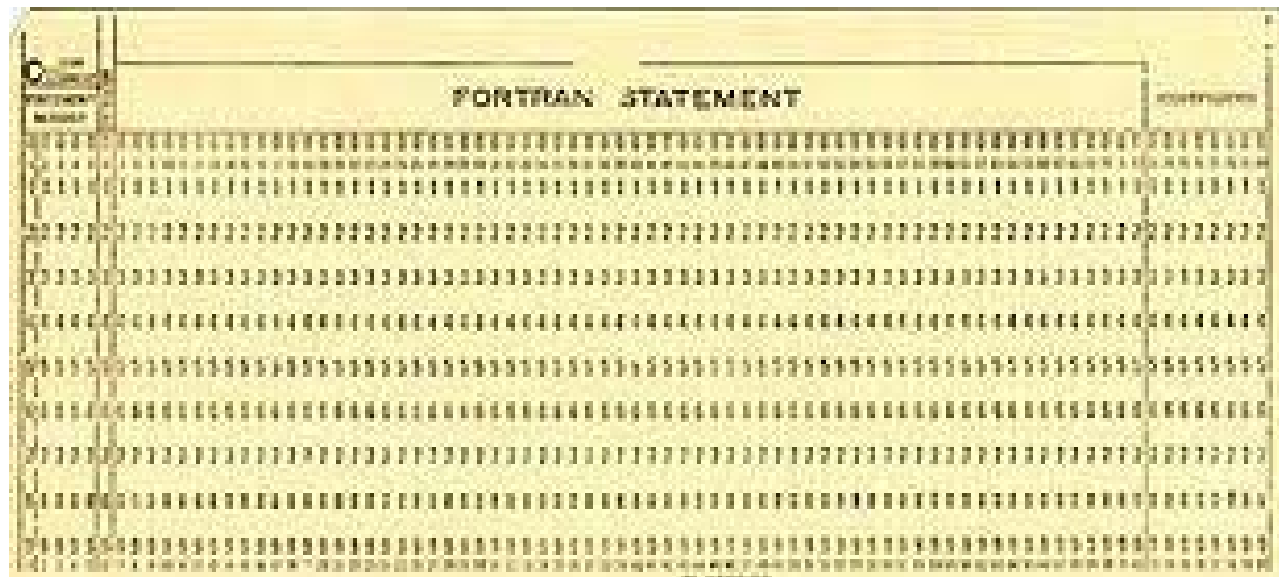
L'informazione è rappresentata dai suoi dati che a loro volta sono espressi sotto forma di simboli.

Un computer memorizza ed elabora informazioni.

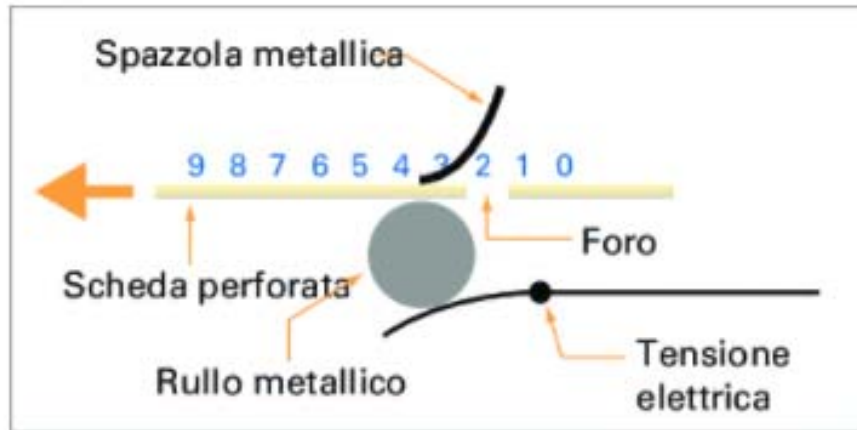
Le informazioni sul calcolatore devono essere rappresentate in una "forma" tale che siano "adatte" ad un certo tipo di memorizzazione e manipolazione

Bit e memorizzazione

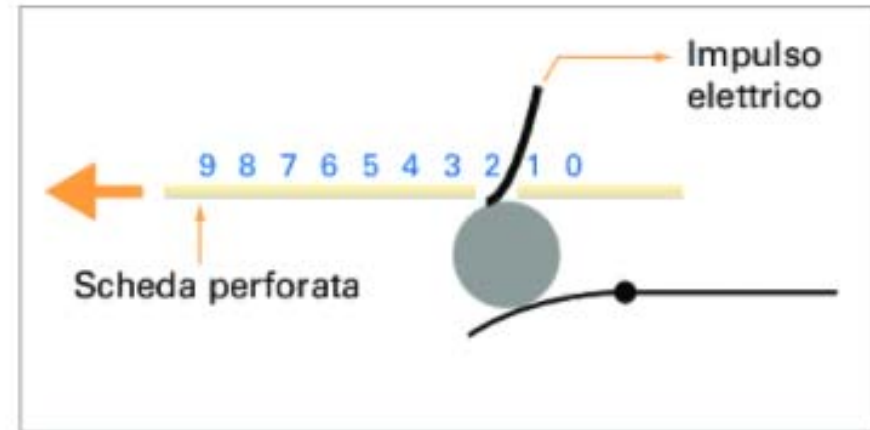
- I computer rappresentano le informazioni attraverso bit (binary digit) che può assumere solo 2 valori 0 e 1.
- Una sequenza di bit può rappresentare numeri, caratteri, immagini ecc.
- Quindi necessariamente per memorizzare un bit occorre un dispositivo che può assumere solo uno dei 2 stati.



Bit e memorizzazione



(a)



(b)

Funzionamento di un lettore di schede perforate. (a) La scheda scorre verso sinistra, mossa da un rullo metallico. (b) Quando un foro passa sopra il rullo, la spazzola lo attraversa e chiude un contatto elettrico.

Digitale

Il termine digitale deriva dall'inglese "DIGIT" (CIFRA) che rappresenta l'elemento della numerazione binaria all'interno di un insieme composto solo da due elementi: "0" e "1".

- circuito aperto – circuito chiuso → elettronica digitale che è alla base dei calcolatori.

Eempio: lampadina collegata ad un interruttore che può avere solo due stati ON e OFF (acceso, spento).

Con 8 lampadine controllate da 8 interruttori si costruisce un byte ovvero una sequenza di 8 cifre binarie dove ciascuna di esse può assumere due e solo 2 valori "1" equivalente ad "ON" e "0" equivalente ad "OFF" e ogni singola cifra è chiamata "bit". Un byte è formato da 8 bit.

Formalizzando: $\text{bit} = \{0,1\}$ e $\text{byte} = \text{bit}_7, \text{bit}_6, \dots, \text{bit}_0$

Codifica e rappresentazione dell'informazione

Ogni sistema di numerazione è costituito da un insieme finito di simboli.

Esistono due tipologie di sistemi di numerazione:

Posizionale e Non posizionale

In un sistema di numerazione posizionale i simboli assumono valori diversi a seconda della posizione che occupano nella notazione.

Viceversa, in un sistema non posizionale il valore attribuito a ciascun simbolo è sempre lo stesso indipendentemente dalla posizione che occupa.

Sistema di Numerazione non posizionale

- Un esempio di numerazione non posizionale è dato dal sistema di numerazione Romano.
In questo sistema additivo/sottrattivo i simboli ammessi sono:

$$\text{Simboli}_{\text{Romano}} = \{I, V, X, L, C, D, M\}$$

con le seguenti equivalenze nel sistema decimale: I = 1; V = 5; X = 10; L = 50; C = 100; D = 500; M = 1 000

N.B. L'insieme dei simboli ammessi è chiamato alfabeto

- Quando si giustappongono più simboli consecutivi (che chiameremo stringa e lunghezza della stringa il numero di simboli) quando il simbolo che precede è maggiore del successivo il valore è ottenuto dalla somma (additivo), quando il successivo è maggiore del precedente il valore è ottenuto dalla differenza (sottrattivo)

*Banalmente il sistema di numerazione non posizionale più intuitivo è quello che usa un solo simbolo: "|"
che ovviamente ha sempre valore di unità qualunque sia la sua posizione: |=1, ||=2, |||=3, ||||=4, ...*

Sistema di numerazione posizionale

I sistemi di numerazione si riferiscono quindi alla successione dei cosiddetti numeri naturali. I più antichi sistemi di numerazione hanno *base dieci*, con riferimento all'atto di contare con le dita delle mani. Per una definizione più formale di sistema di numerazione posizionale:

- si sceglie un qualsiasi numero naturale b (diverso da zero e da uno), che chiameremo base;
- si scelgono m simboli diversi
- si compongono i numeri tenendo presente che il valore di ogni cifra va moltiplicato per:
 - b^0 cioè 1 (unità) se è l'ultima cifra alla destra del numero che stiamo considerando
 - b^1 cioè b se è la seconda cifra
 - b^2 se è la terza cifra
 - e così via, $b^{(n-1)}$ se è la n -esima cifra
- la somma di tutti i valori così ottenuti è il numero che stiamo considerando

Sistema di numerazione posizionale

Il sistema di numerazione posizionale necessita dello zero per occupare con valore nullo la posizione della cifra che occupa all'interno della stringa

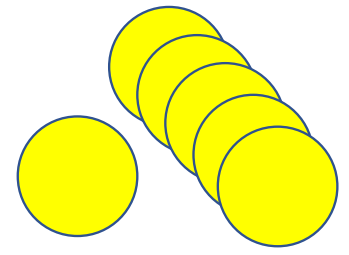
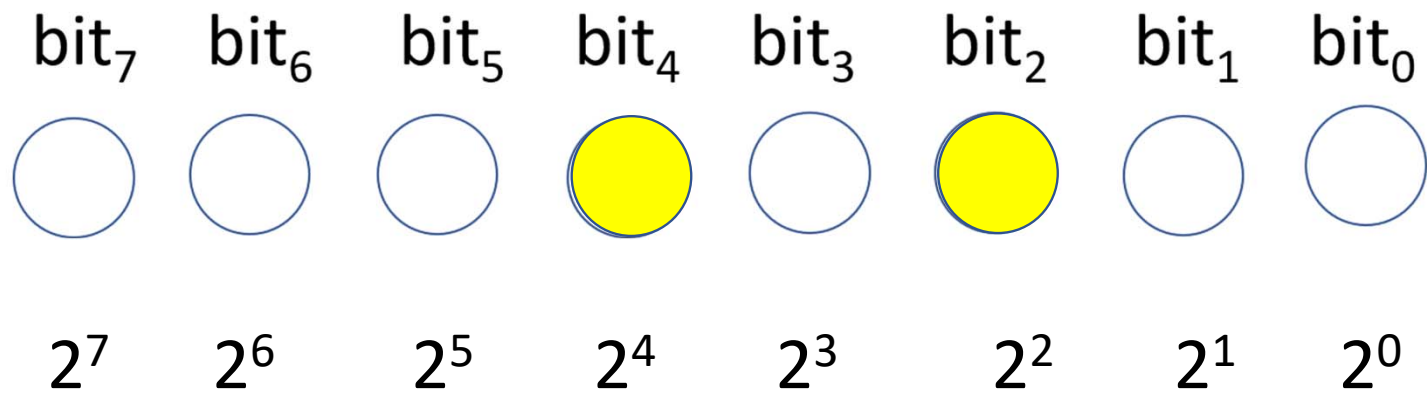
- Ciascun alfabeto è composto da una quantità finita di simboli.
- Il sistema decimale è costruito su 10 simboli
- Il sistema binario è costruito su 2 simboli
- Il sistema esadecimale è costruito su 16 simboli

Sistema di numerazione posizionale

Le ragioni della superiorità del sistema numerico decimale-posizionale, che si è diffuso dall'India, sono il principio posizionale (che di per sé denota i diversi ordini numerici) e l'uso di dieci simboli, comprensivi dello zero (che colmava i vuoti in un sistema posizionale).

Il sistema decimale-posizionale consente anche una comoda esecuzione di operazioni aritmetiche

Byte



Rappresentazione in base 2

Base binaria: $p=2$; simboli $m \in \{0, 1\}$

Ogni singola cifra è chiamata bit (binary digit)

In base $p=2$, simboli $m=2$, numero di cifre $n=4$

posso rappresentare numeri nell'intervallo $[0, 2^4-1]=[0, 15]$

Ovvero: $[0000, 1111]$

Ad esempio: $1010_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 1^0 = 8 + 0 + 2 + 0 = 10_{10}$

Simboli

Simboli₂ $\in\{0, 1\}$

Num. di Simboli=2

Simboli₁₀ $\in\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

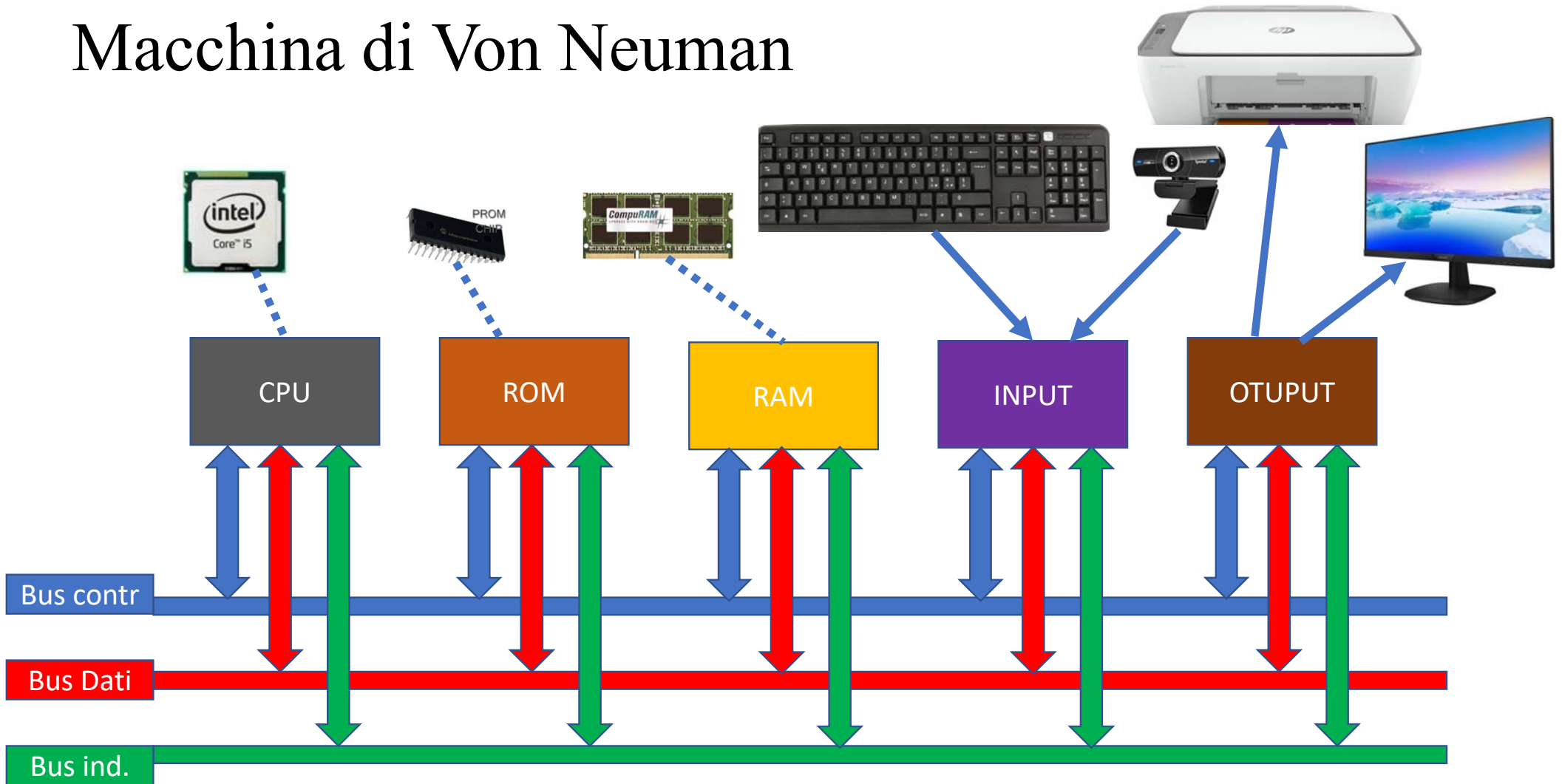
Num. di Simboli=10

Simboli₁₆ $\in\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

Num. di Simboli=16

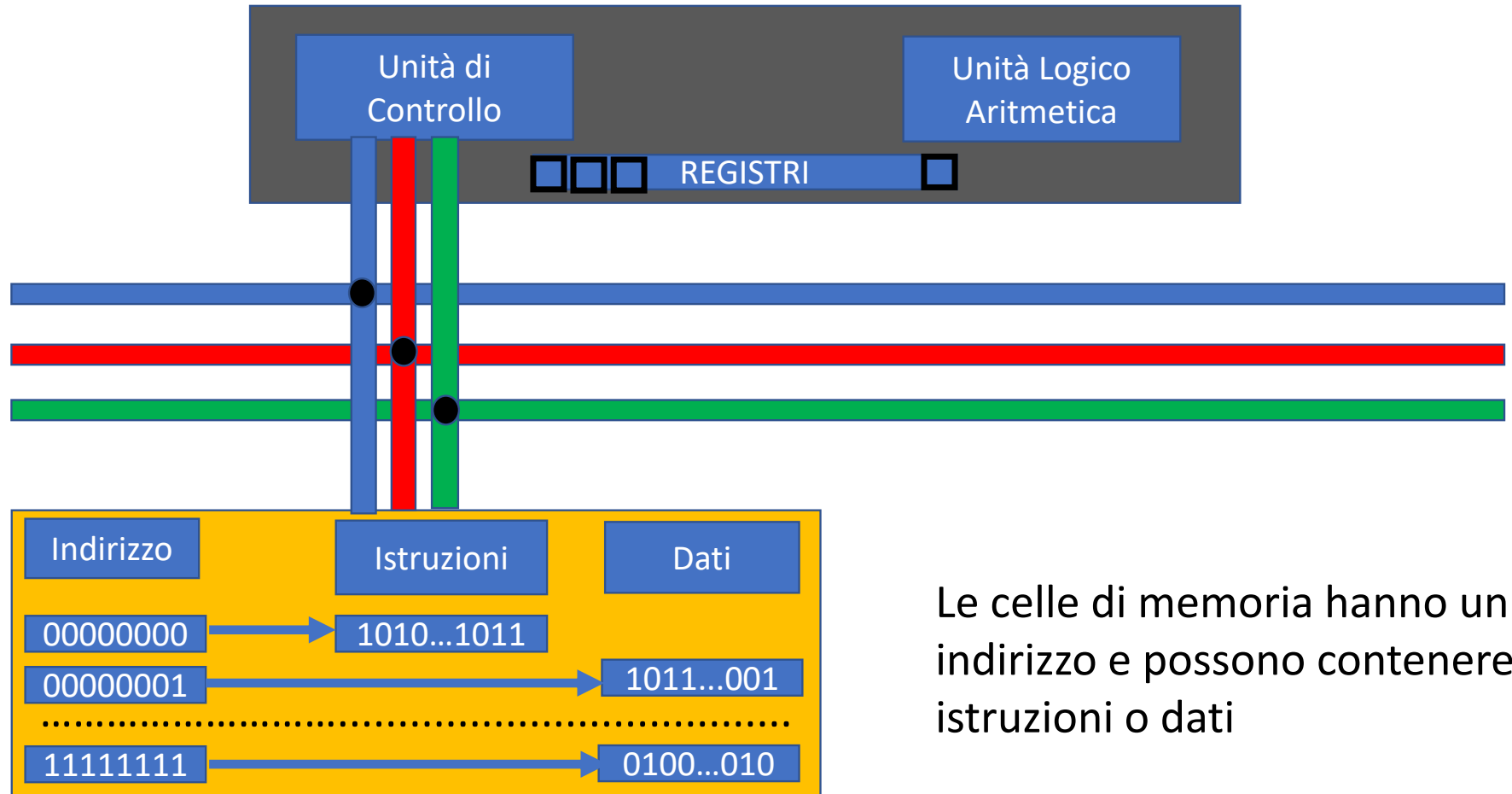
$1010_{16} =$

Macchina di Von Neuman



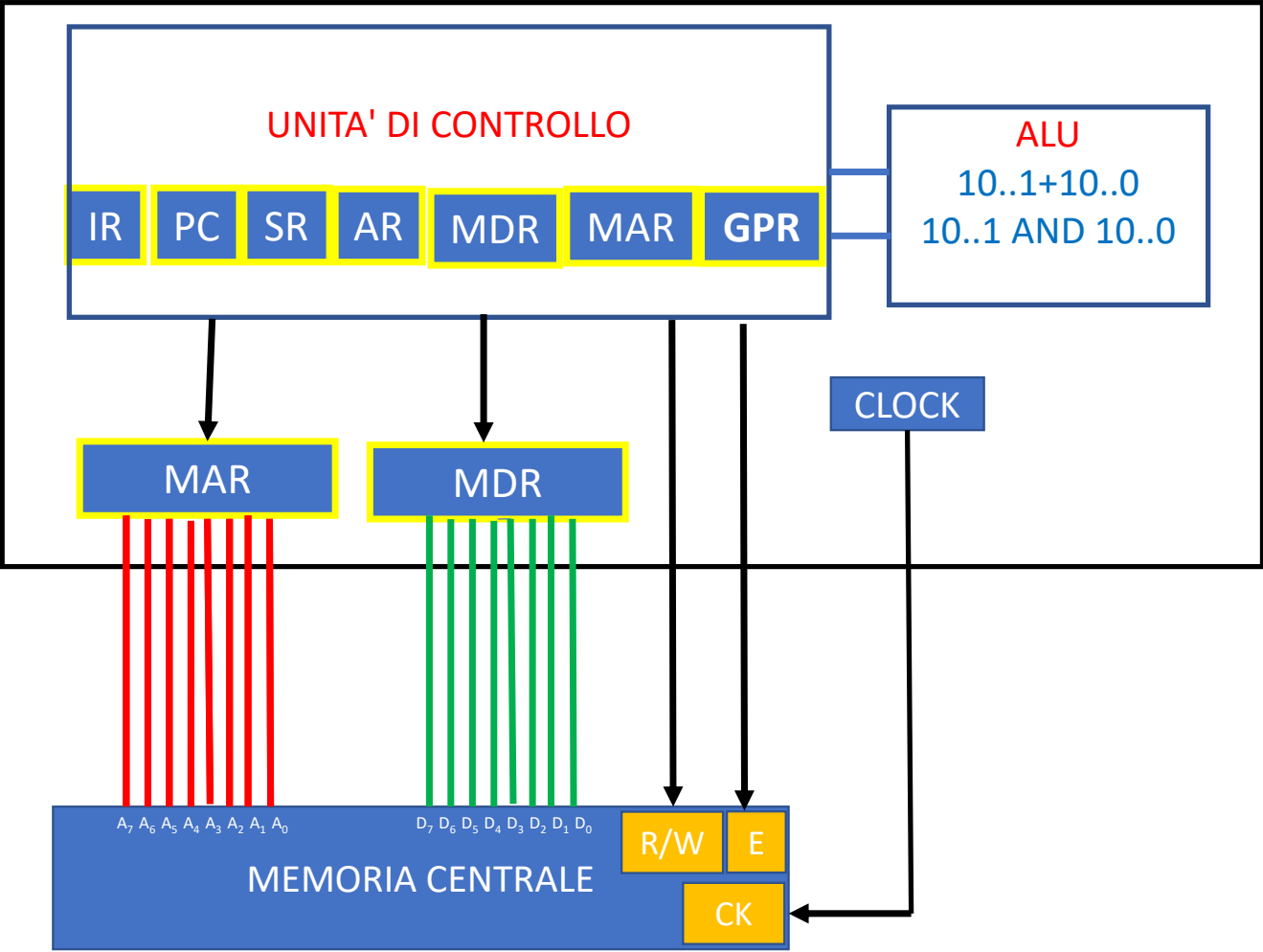
Processore - Memoria Centrale - Unità di I/O - Bus di sistema

MEMORIA



Le celle di memoria hanno un indirizzo e possono contenere istruzioni o dati

Registri della CPU



- PC = Program Counter**
- IR = Instruction Register**
- SR = Status Register**
- AR = Accumulator Register**
- MDR = Memory Data Register**
- MAR = Memory Address Register**
- GPR = General Purpose Register**

Indirizzamento della memoria

- Il Memory Address Register (MAR) contiene l'indirizzo della cella di memoria che deve essere selezionata.
- La lunghezza in bit (n) del MAR determina la quantità massima di celle di memoria indirizzabili. Queste vanno da "0" a 2^n-1 .

Es per 16 bit $\rightarrow 2^{16}-1 = 65535$ ovvero sono indirizzabili 65535 celle.

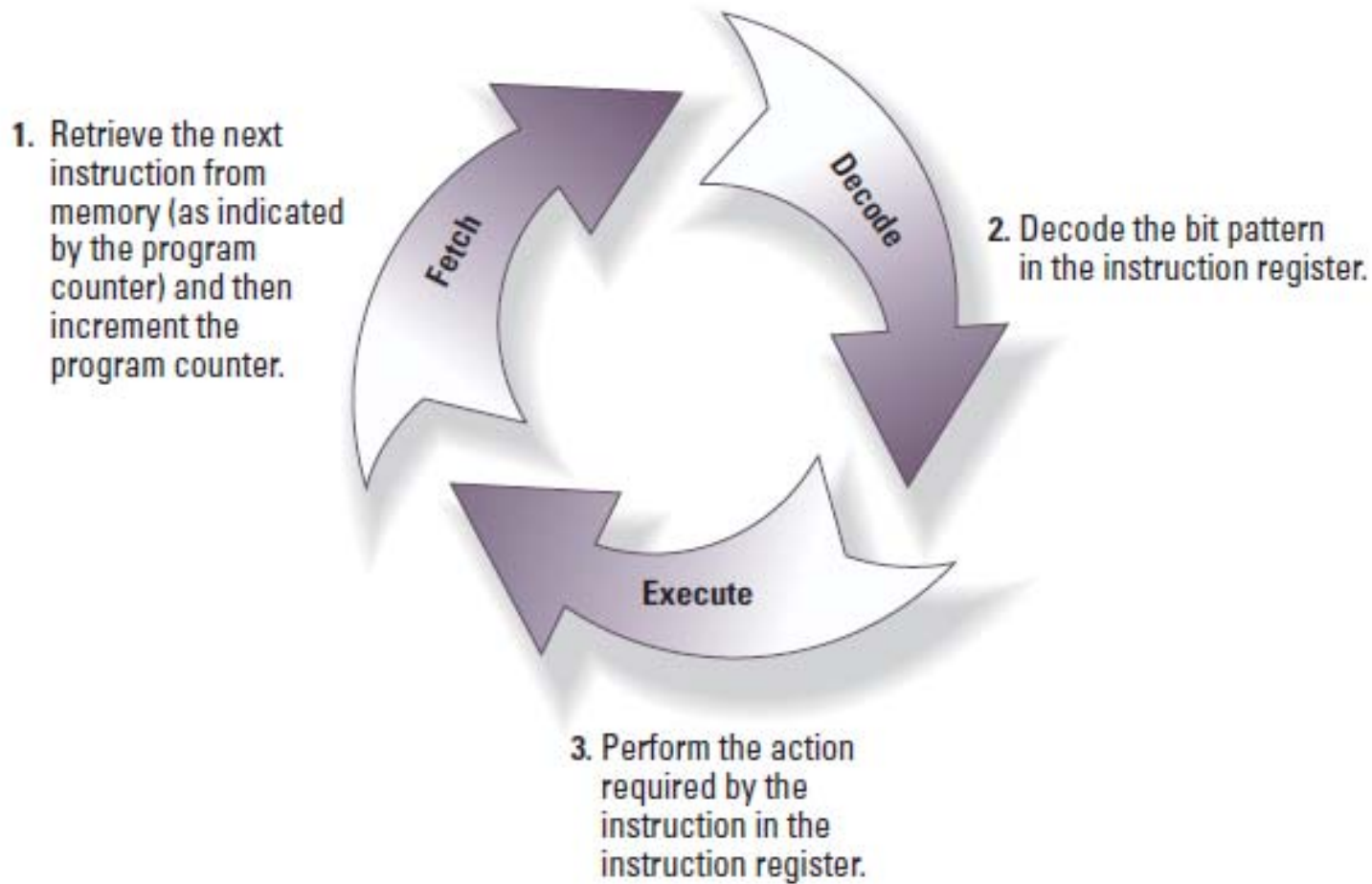
Unità di Controllo

- Questa unità controlla e determina il funzionamento del computer ed esegue ciclicamente tre FASI in maniera consecutiva:
- Fase di FETCH (prelievo)
- Fase di DECODE (decodifica)
- Fase di EXECUTE (esecuzione)

CICLO di Fetch-Decode-Execute

- **Fetch** (prelievo)
 1. accedi alla prossima istruzione (vai all'indirizzo della memoria in cui è contenuta la nuova istruzione)
 2. Memorizza l'indirizzo nel registro IR
- **Decode** (decodifica)
 1. Decodifica l'istruzione in base al suo codice operativo (che può essere una operazione di somma ... ecc)
- **Execute** (esecuzione)
 1. Si individuano i dati usati nell'istruzione
 2. Si trasferiscono i dati nei registri opportuni
 3. Si esegue l'istruzione

CICLO di Fetch-Decode-Execute



Memoria Cache

- All'interno della CPU esiste questa ulteriore memoria (di piccole dimensioni) che funziona alla velocità del processore (molto superiore a quella delle RAM).

Calcolo parallelo

Le operazioni della macchina di Von Neumann sono sequenziali, i nuovi calcolatori utilizzano più processori in tal modo possono eseguire operazioni contemporaneamente velocizzando i processi di calcolo.

Rappresentazione dell'informazione

Digitalizzare significa rappresentare l'informazione per mezzo di cifre.

Ad esempio per identificare e rappresentare una persona usiamo il C.F., per una azienda la P.IVA, per un prodotto commerciale il codice a barre.

Per la digitalizzazione abbiamo bisogno di usare un insieme di simboli che possono essere numerici, ma anche simboli speciali ;
{*. %.@ ecc.}

La tastiera, codifica ASCII

Quando digitiamo un tasto, inviamo al calcolatore, tramite un collegamento seriale, una sequenza di bit che definisce in modo univoco quel determinato simbolo.

Quanti sono i bit necessari per codificare tutti i simboli di una tastiera (compresi gli spazi, i numeri, gli a capo, le tabulazioni ecc.)?

inizialmente con la prima codifica ASCII (Standard Code for Information Interchange) venivano utilizzati $2^7 = 128$ combinazioni, successivamente si è passati, per codificare tutti i caratteri di tutte le lingue, ad una codifica ad 8 bit = 256 combinazioni chiamata Extended ASCII.

Codifica Extended ASCII

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

Quindi ad esempio il digitando la lettera "a" su una tastiera la sua codifica è 01100001 oppure uno spazio con 00100000

Byte	Cod.	Char
01100000	96	`
01100001	97	a

Byte	Cod.	Char
00100000	32	Spc
.....		

Algoritmi

- L'algoritmo è un metodo preciso e sistematico per produrre un determinato risultato.
- Un processo può essere definito algoritmo se ha 5 proprietà:
 1. Input specificato → quali sono i dati forniti all'elaborazione
 2. Output specificato → quali i dati che questo produce
 3. Determinatezza → tutto l'intero processo non deve essere non ambiguo
 4. Efficacia → **l'agente** deve essere in grado di eseguire le istruzioni fornite
 5. Terminazione → l'agente prima o poi dovrà alla fine fornire una risposta corretta

Esempio di algoritmo di ricerca lineare

Problema: vogliamo cercare un nome di uno studente in una classe.

Dati di INPUT: un elenco degli studenti della classe

Vogliamo che il risultato di OUPUT sia:

A) "Si, ho trovato il nome" oppure B)"No, non ho trovato il nome"

1. Colloca un contrassegno sul primo elemento
 2. Se l'elemento su cui si trova il contrassegno non è quello cercato invia il messaggio **B**
 3. Se l'elemento su cui si trova il contrassegno è quello cercato invia il messaggio **A**
 4. Avanza il contrassegno alla posizione successiva
 5. Torna al passo 2
- Algoritmi e processi euristici
 - escamotage mentali che portano a conclusioni veloci con il minimo sforzo cognitivo.

Macchina di Turing

Ancora oggi questo modello viene utilizzato come strumento per analizzare la potenza dei processi algoritmici.

Intorno al 1930, ad opera di matematici e logici quali J. Herbrand, K. Godei, S.C. Kleene, E.L. Post, A.M. Turing, si è sviluppata la teoria generale degli algoritmi, che ha portato a precisare rigorosamente il concetto di computabilità e a stabilire con esattezza quali funzioni sono computabili, cioè risolvibili mediante algoritmi, e quali n