

# Fondamenti di Informatica

Accademia di Belle Arti di Verona

Università degli Studi di Verona

A.A. 2016-2017

Docente - Vincenzo Giannotti

# CAPITOLO 2 – ALGORITMI E COMPUTER

# Il concetto di Algoritmo

- Nella nostra carrellata storica abbiamo visto che molti hanno tentato per secoli di costruire delle macchine automatiche in grado di realizzare delle operazioni e di produrre dei risultati senza l'intervento dell'uomo: da Pascal e Poleni a Babbage e poi Turing, Zuse fino ai grandi scienziati e imprenditori moderni.
- Fare delle operazioni in maniera automatica significa: realizzare degli **algoritmi** senza che sia dovuto l'intervento dell'uomo.
- Poiché questo è l'unico scopo di una macchina automatica e poiché noi siamo interessati a capire come funziona una tale macchina, sarà utile, prima di proseguire, cercare di comprendere bene cosa sia un algoritmo.

# Il concetto di Algoritmo

- Il dizionario Treccani lo definisce più o meno così:

*“Termine, derivato dall’appellativo al-Khuwārizmī del matematico Muḥammad ibn Mūsa del 9° sec., originario di quello che è l’attuale Uzbekistan, che designa qualunque schema o procedimento sistematico di calcolo. Con un algoritmo si tende a esprimere in termini matematicamente precisi il concetto di procedura generale, di metodo sistematico valido per la soluzione di una certa classe di problemi”.*

- Se questa definizione non è abbastanza chiara, proviamo più semplicemente a dirlo così:

**“un algoritmo è una sequenza finita di operazioni da svolgere per risolvere un dato problema”.**

# Il concetto di Algoritmo

- Perché vi ho presentato queste due definizioni, così diverse, di algoritmo?
- Perché ancora oggi non esiste una «definizione di algoritmo» che ne definisca pienamente il concetto in termini formali ci si deve accontentare di definizioni più o meno intuitive come quella prima evidenziata.
- Uno dei tentativi più riusciti è quello proposto da Turing con la sua «macchina» ed è proprio per questo motivo che la Macchina di Turing è ancora oggi lo strumento più potente utilizzato per decidere se un dato procedimento sia o meno calcolabile con una macchina automatica.

# Il concetto di Algoritmo

- Detto questo, quello di «**Algoritmo**» è forse il più importante concetto dell'Informatica e noi, per comodità, lo considereremo nella sua accezione più semplice che ribadiamo:

**“un algoritmo è una sequenza finita di operazioni da svolgere per risolvere un dato problema”.**

- Se qualcuno vi insegna la ricetta della pasta frolla, vi sta fornendo un algoritmo:
  - In una terrina mettete la farina
  - formate una conca in mezzo ad essa e rompete al centro le uova
  - ora metteteci lo zucchero, la scorza di limone grattugiata, il pizzico di sale e il burro
  - quindi iniziate a lavorare gli ingredienti con la punta delle dita di modo che l'impasto non si riscaldi
  - ad operazione conclusa formate una palla, avvolgetela con la pellicola trasparente per alimenti e adagiatela in frigorifero a riposare per 30 minuti.
- Se l'algoritmo è corretto e voi lo svolgete rigorosamente, certamente la vostra pasta frolla sarà ottima.

# Il concetto di Algoritmo

- Anche senza bisogno che qualcuno ce lo spieghi, vediamo subito che per essere sicuri di poter fare un'ottima pasta frolla abbiamo bisogno che:
  1. Ci sia dato il materiale con cui lavorare (gli ingredienti)
  2. Ci sia spiegato quale procedimento applicare attraverso un numero finito di istruzioni elementari (la ricetta)
  3. Tutto questo, alla fine, produce un risultato (la pasta frolla).
- Utilizzando un linguaggio più affine all'informatica, abbiamo bisogno:
  1. Di un **input**
  2. Di un **algoritmo**
  3. E noi, alla fine, dobbiamo produrre un **output**.

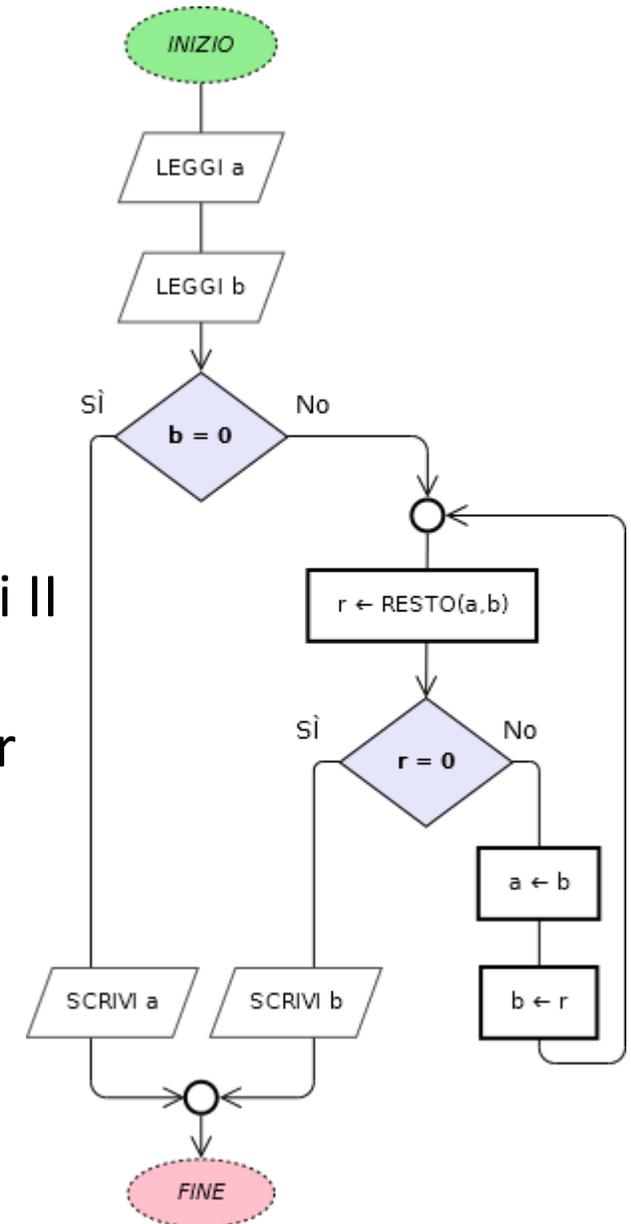
# Il concetto di Algoritmo

Classici esempi di algoritmi in informatica sono:

- L'algoritmo di Euclide per il calcolo del MCD di due numeri interi
- L'algoritmo per la ricerca delle soluzioni di una equazione di II grado
- L'algoritmo *PageRank di Google*, che stabilisce quale sito far apparire prima nei risultati di una ricerca

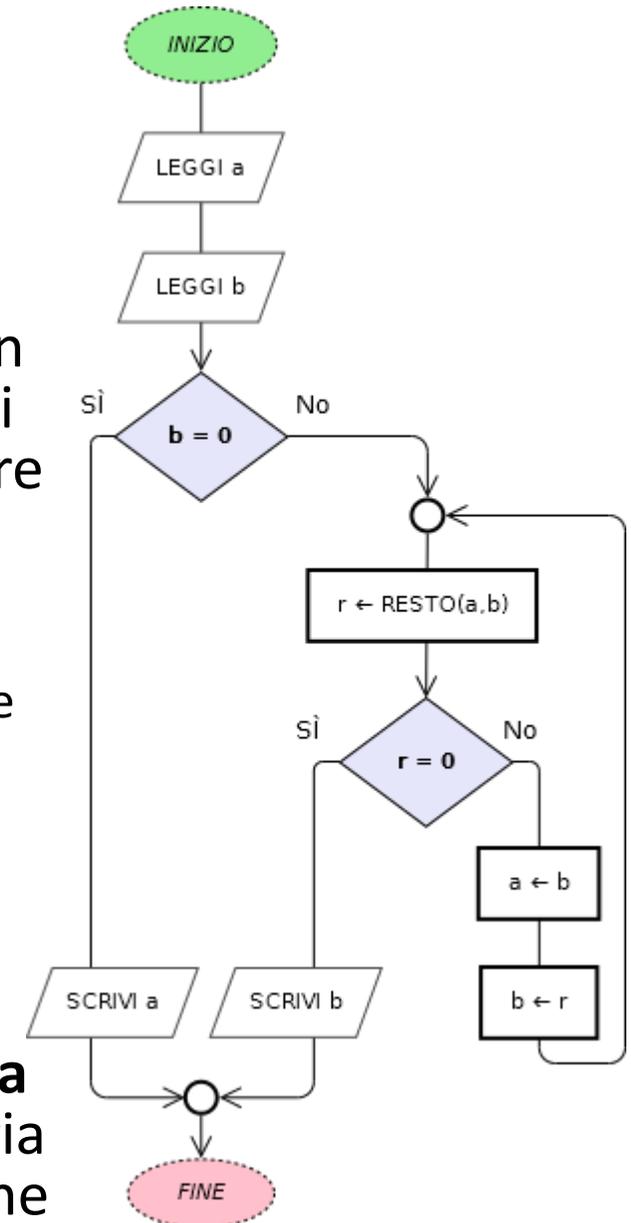
Ma anche i seguenti sono esempi di algoritmi:

- Le indicazioni stradali per raggiungere una data località
- Le istruzioni per assemblare un mobile dell'IKEA



# I diagrammi di flusso

- In informatica il diagramma di flusso (in inglese **flow chart**) è un linguaggio di modellazione grafico per rappresentare il flusso di controllo ed esecuzione di algoritmi; esso consente di descrivere in modo schematico ovvero grafico:
  - le operazioni da compiere, rappresentate mediante sagome convenzionali (rettangoli, rombi, esagoni, parallelogrammi, rettangoli smussati...), ciascuna con un preciso significato logico e all'interno delle quali un'indicazione testuale descrive tipicamente l'attività da svolgere
  - la sequenza nella quale devono essere compiute, rappresentate con frecce di collegamento (flusso).
- Per questa loro connotazione topologica i diagrammi di flusso possono essere ricondotti alla classe più ampia dei **diagrammi a blocchi**, che a loro volta rientrano nell'ancora più vasta categoria delle **mappe concettuali**, utilizzate soprattutto per la descrizione e la rappresentazione delle informazioni e della conoscenza.



# Proprietà degli Algoritmi

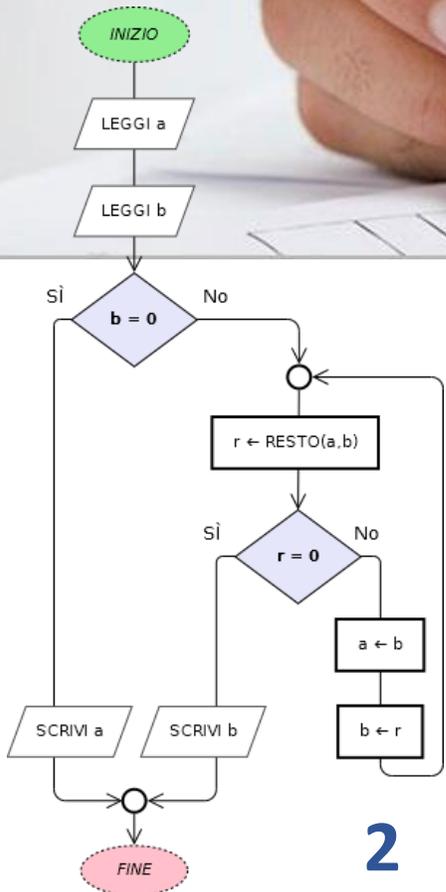
- Abbiamo visto che quello di algoritmo non è un concetto strettamente informatico.
- In Informatica tuttavia, per essere definito tale, un algoritmo deve rispettare alcune proprietà, definite negli anni '60 da Donald Knuth (Stanford University):
  - **Finitezza**: Un algoritmo deve terminare dopo un numero finito di passi
  - **Definitezza**: Ogni passo deve essere rigorosamente definito; le azioni da svolgere devono essere rigorose e non ambigue (operazioni elementari)
  - **Input**: ...deve avere un numero finito di dati/oggetti in entrata
  - **Output**: ...deve produrre un risultato univoco
  - **Efficacia**: Tutte le operazioni da svolgere debbono avvenire in un tempo finito ragionevole
- Queste proprietà servono a garantire il fatto che un computer sia in grado di portare a termine il suo compito.

# Dall'algoritmo al programma

- Prima che una macchina come il computer possa realizzare un dato compito, è necessario che sia concepito e definito un algoritmo o una serie di algoritmi in grado di realizzare quello stesso compito («**analisi**»)
- In seguito quello stesso algoritmo dovrà essere rappresentato in una forma che sia compatibile con il nostro computer; viene fatto un «**progetto**» per trascrivere l'algoritmo utilizzando un linguaggio che la macchina sia in grado di comprendere
- Questa rappresentazione dell'algoritmo si chiama «**programma**» e l'intero processo di sviluppo del programma, la sua codifica in forma comprensibile alla macchina e l'inserimento in memoria, viene detto «**programmazione**»
- Il programma viene verificato in una fase di «**testing**» in cui le funzionalità del programma vengono valutate (rispetto alle specifiche) ed il programma viene utilizzato da persone diverse per individuarne eventuali malfunzionamenti
- Infine si ha l'attività di «**manutenzione**» per ulteriori correzioni, adattamenti a diverse piattaforme, evoluzioni etc.

# Dall'algoritmo al programma

1



2

```
PROGRAM Euclide;  
VAR  
    M,N:integer;  
  
FUNCTION MCD(M,N:integer):integer;  
    VAR  
        Resto:integer;  
  
    BEGIN  
        Resto:=M MOD N;  
        WHILE Resto<>0 DO  
            BEGIN  
                M:=N;  
                N:=Resto;  
                Resto:=M MOD N  
  
            END;  
        MCD:=N  
  
    END;  
  
BEGIN  
    writeln('Assegna M e N ');  
    read(M);read(N);  
    writeln('Il MCD e'':= ',MCD(M,N));  
    readln  
END.
```

3



4

# Dall'algoritmo al programma

- In linea teorica, una volta che viene scoperto un nuovo algoritmo che realizza un dato compito (per esempio l'algoritmo di Euclide per trovare il Massimo Comun Divisore di due numeri interi, l'algoritmo per comprimere la musica in mp3 o le immagini in jpg), questo viene dato per acquisito e non viene richiesto di comprenderne il funzionamento.
- In questo senso l'algoritmo rappresenta un «mattoncino fondamentale» che non viene ulteriormente messo in discussione.
- Soprattutto nell'informatica moderna, questo concetto è ampiamente utilizzato e un grandissimo numero di questi «mattoncini» si trovano nel Web ad uso e consumo dei programmatori e degli appassionati di tutto il mondo.

# Algoritmi e Società

Oggi l'informatica si è affermata come la scienza degli algoritmi. L'ambito di questa scienza è molto ampio, attingendo e interagendo con diverse discipline come la matematica, l'ingegneria, la psicologia, la biologia, l'economia aziendale, le scienze ambientali, la linguistica etc... Questo implica che i ricercatori in diversi rami dell'informatica possono avere delle idee anche molto diverse di cosa sia l'informatica e di quale sia il suo scopo.\*

# Algoritmi e Società

Ad esempio, un ricercatore che lavora nel campo delle architetture di computer può avere in mente che la scienza dell'informatica sia legata per lo più all'**avanzamento tecnologico**; un ricercatore nel campo dei database può vedere informatica come la ricerca di modi sempre più efficienti per archiviare e rendere l'**informazione accessibile** e contribuire ad una società più consapevole e trasparente; un ricercatore nel campo delle applicazioni per l'ambiente può pensare di poter contribuire ad un mondo più pulito e uno nel campo dell'intelligenza artificiale può considerare l'informatica legata allo studio dell'intelligenza umana e dei **comportamenti**, immaginando società future dove le macchine avranno un ruolo preponderante.\*

\*Liberamente tratto da «Computer Science: an overview 11° ed» - j. Brookshear (Pearson 2012)

# Algoritmi e Società

- Queste sono visioni parziali in cui si perde di vista il quadro generale e si perde di vista il ruolo vero dell'informatica che non è quello di cambiare il mondo quanto piuttosto quello di rispondere ad alcuni quesiti (in quanto scienza degli algoritmi).
  - Quali problemi possono essere risolti con un approccio di tipo algoritmico?
  - Come può essere reso più semplice scoprire e definire un algoritmo?
  - Come possono essere migliorate le tecniche di rappresentazione e di comunicazione degli algoritmi?
  - Come possono essere analizzate e comparate le caratteristiche di diversi algoritmi?
  - Come possiamo utilizzare gli algoritmi per manipolare l'informazione?
  - Come possiamo applicare degli algoritmi per produrre comportamenti intelligenti?
  - Che impatto può avere l'applicazione degli algoritmi sulla società?
- Rispondere a queste domande presuppone di operare con un elevato grado di astrazione il che significa: ignorare i dettagli interni di un dato oggetto, per concentrarsi sulle sue proprietà esterne\*.

# Il Computer

Ripartiamo dalla definizione di Computer:

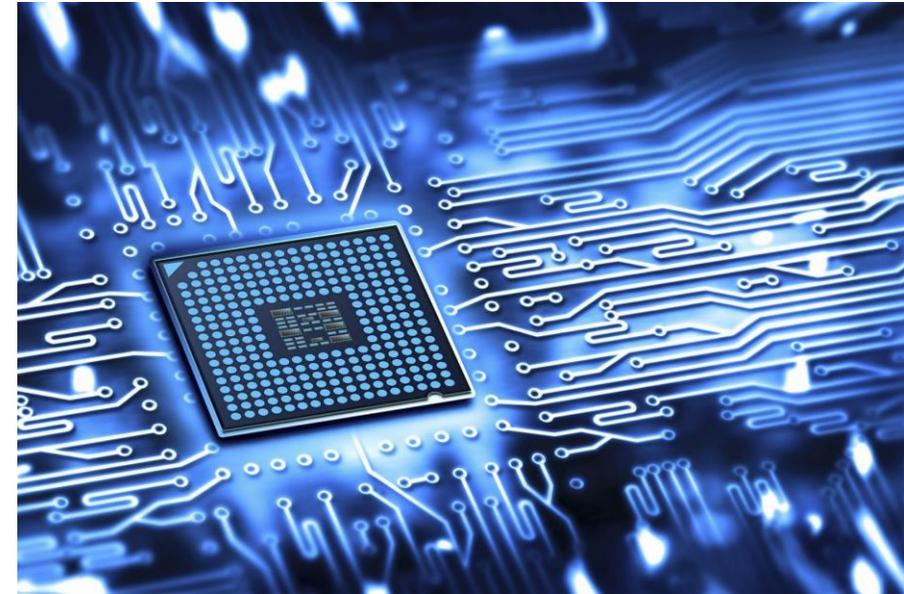
**«Macchina in grado di elaborare dati, automatica e programmabile»**

e dalle sue due componenti fondamentali:

- La componente fisica ovvero «Hardware»;
- La componente immateriale ovvero «Software»



# Hardware



# Software



## Google Algorithm



# Il Computer

La prima cosa da fare per cercare di spiegare il computer è quella di porsi la seguente domanda:

**Perché il computer è diventato uno strumento così importante e pervasivo della nostra società?**



# Il Computer



Immagini, testi, voce, dati, film, musica, vengono codificati in sequenze di BIT  
0110011100110...

**Perché tutta l'informazione del pianeta può essere tradotta attraverso sequenze di impulsi elettrici.**



# Il Computer

- A differenza della mente umana che è in grado di affrontare e risolvere problemi nuovi utilizzando capacità cognitive, intuizione e ragionamento, il computer, che è una macchina, è invece un semplice «**esecutore**» di istruzioni impartite dall'esterno.
- Le istruzioni vengono impartite attraverso un algoritmo studiato per risolvere un problema a partire da dei dati di input affinché si produca un risultato in output, attraverso l'elaborazione dei dati stessi.
- Il computer, in quanto mero esecutore di algoritmi, deve essere opportunamente istruito e ciò avviene ad opera di un programmatore capace di comunicare e interagire con la macchina attraverso dei **linguaggi di programmazione** che possono essere ad alto, medio o basso livello di astrazione.
- Il più basso livello di astrazione è il cosiddetto «**linguaggio macchina**».

# Il Computer

A questo punto potremmo porci delle nuove domande:

- Cosa sono i BIT e cosa è la codifica digitale?
- Cos'è un programma e come lo si rappresenta?
- Cosa è che fa lavorare il computer?
- Che cosa ne governa il funzionamento?
- Come possono i computer dialogare fra di loro?
- Quali rischi corriamo quando utilizziamo un computer?

A tutto questo cercheremo di dare una risposta.

# BIT e BYTE

Abbiamo già visto che i computer hanno bisogno di un proprio linguaggio per poter comunicare tra loro e con gli esseri umani.

Così come gli uomini utilizzano dei simboli elementari nel proprio linguaggio (fonemi per linguaggio verbale; caratteri per linguaggio scritto), anche il computer utilizza delle proprie entità elementari che noi codifichiamo in:

**0 e 1**

**BIT (Binary digIT)**

Tutta l'informatica è costruita su questi due simboli.

# BIT e BYTE

## Perché vengono utilizzati solo i due simboli 0 e 1 ?

La ragione è già detta e facilmente intuibile: dal punto di vista tecnologico è molto facile rilevare il passaggio o meno di una corrente all'interno di un conduttore. Sarebbe altresì molto difficile e dispendioso costruire dei circuiti in grado di distinguere tra dieci segnali diversi (0..9).

Quella binaria invece è una «informazione di stato» molto semplice che può assumere soltanto due valori: vero/falso; acceso/spento; si/no. In pratica:

- La corrente c'è  $\Rightarrow 1$
- La corrente non c'è  $\Rightarrow 0$

Attraverso diverse combinazioni di 1 e 0 può essere rappresentato, in un computer, l'intero «**Universo delle Informazioni**».

# BIT e BYTE

Il BIT è l'elemento fondamentale di informazione ma normalmente viene aggregato in un insieme che costituisce esso stesso un elemento importante in informatica. Questo è il

## BYTE

Il BYTE è costituito da una sequenza di 8 BIT. Poiché ad un BYTE può essere associato uno dei nostri caratteri, esso talvolta viene anche detto «carattere».

Il BYTE è anche l'entità elementare utilizzata nella memorizzazione.

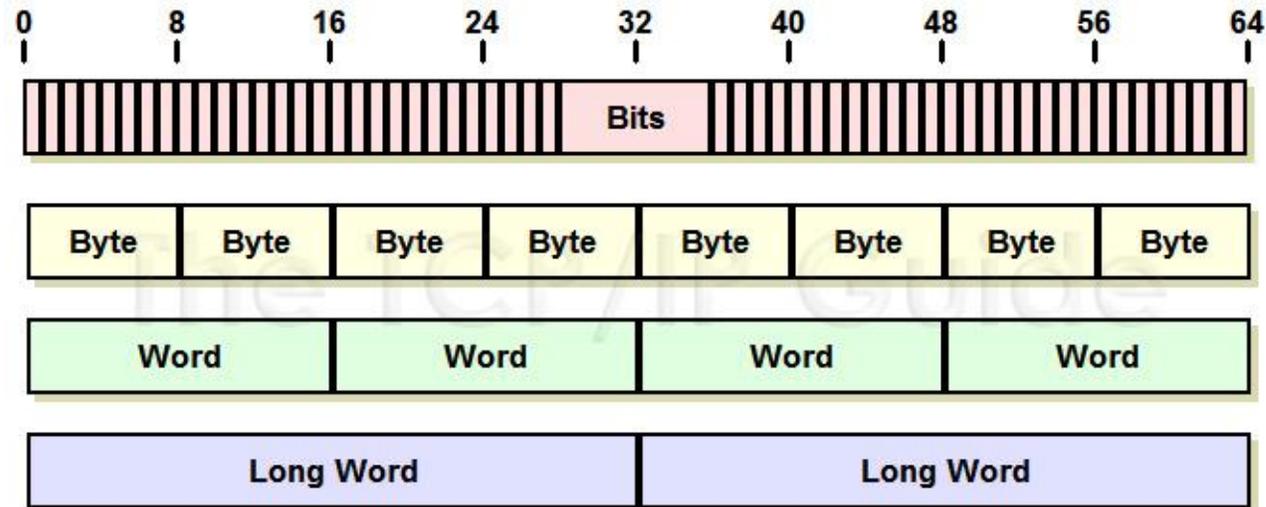
ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
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

# BIT e BYTE

Un altro termine importante da conoscere e che spesso viene utilizzato come riferimento è «WORD» che è dato da 2 BYTE e LONGWORD (4 BYTE).

Con WORD si rappresentano per esempio i piccoli interi, con LONGWORD gli interi e i numeri reali, e così via.



# I multipli del BIT e del BYTE

È importante a questo punto fare una breve panoramica su quelli che sono i «multipli» del BIT/BYTE che ci servono per rappresentare, per esempio, la capacità di memoria o la velocità di un dispositivo di archiviazione.

- KILO **k** ≈ un migliaio Kb (Kilobit) – KB (Kilobyte)
- MEGA **M** ≈ un milione Mb (Megabit) – MB (Megabyte)
- GIGA **G** ≈ un miliardo Gb (Gigabit) – GB (Gigabyte)
- TERA **T** ≈ mille miliardi Tb (Terabit) – TB (Terabyte)

■ 1,024 (e.g. one Kilobyte = 1,024 bytes).

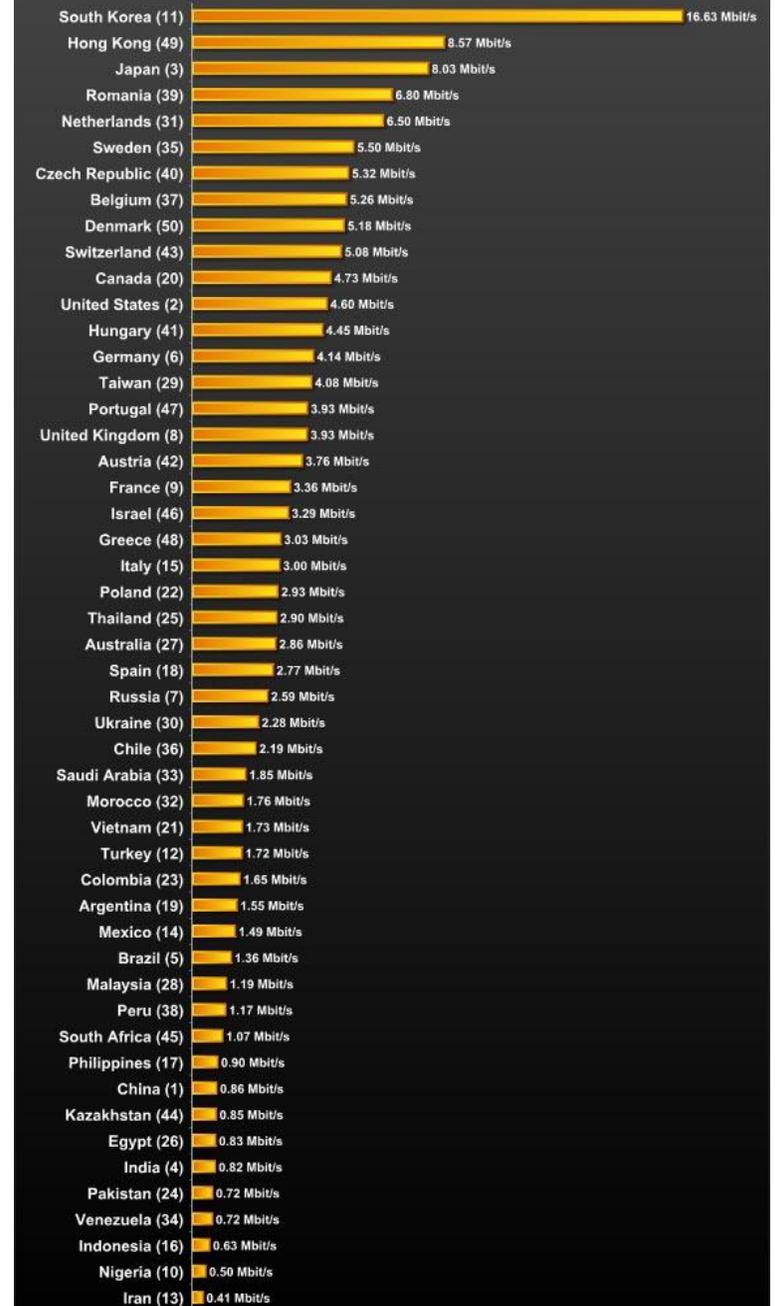
- Bit = 1 bit
- Byte = 8 bits
- Kilobyte = 1024 bytes
- Megabyte = 1024 kilobytes
- Gigabyte = 1024 megabytes
- Terabyte = 1024 gigabytes
- Petabyte = 1,048,576 gigabytes
- Exabyte = 1,073,741,824 gigabytes
- Zettabyte = 1,099,511,627,776 gigabytes

# BIT e BYTE

È importante sottolineare che i valori dati (come potenze di 2) valgono quando si parla di supporti di memorizzazione, non nel caso ci si riferisca a questi valori per misurare la velocità di trasmissione.

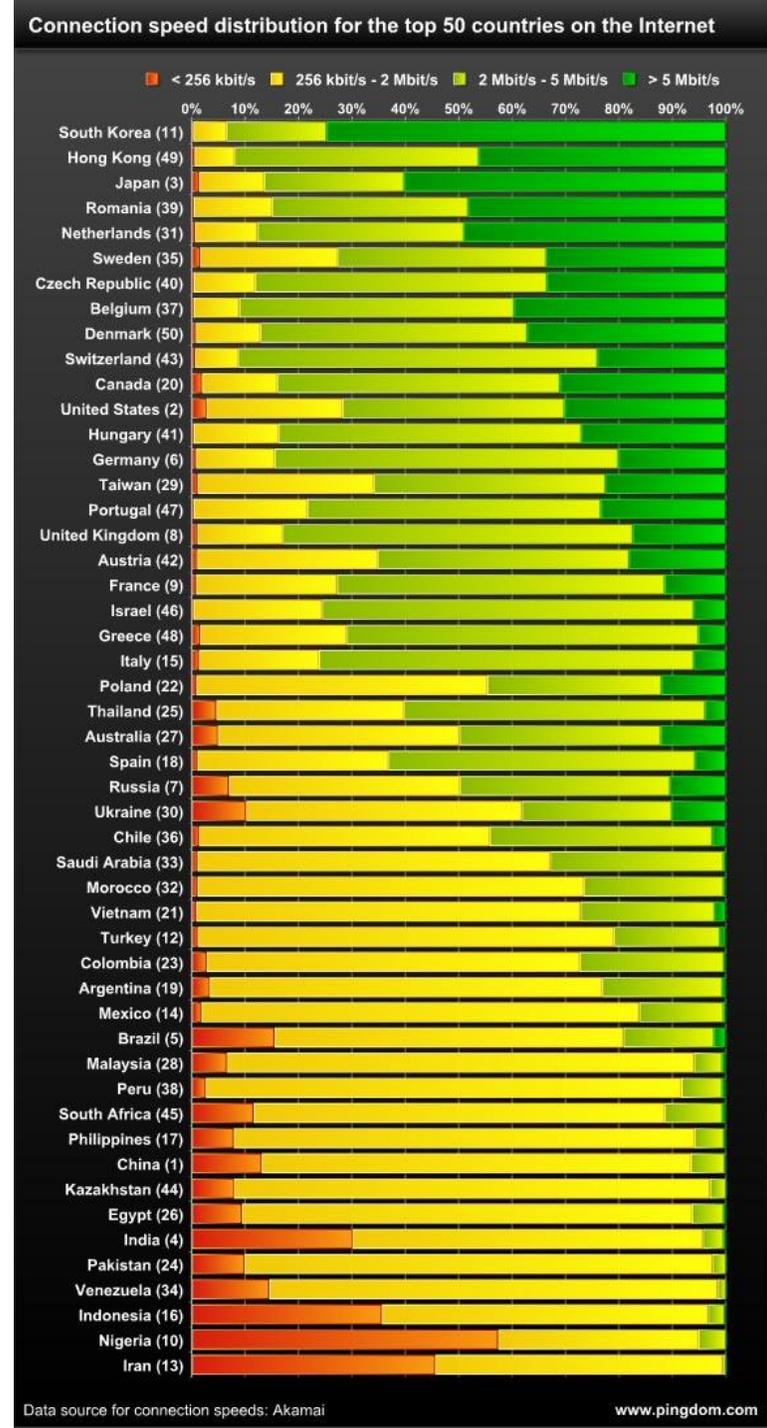
Per esempio, quando vi interconnettete a Internet e vi viene segnalato che state scaricando un file alla velocità di 300Kb/s, sappiate che in questo caso 300K vale 300.000 e non 307.200 (300X1024).

Average connection speeds for the top 50 countries on the Internet



# BIT e BYTE

- La velocità media di connessione per gli utenti di Internet in tutto il mondo, non solo tra questi 50 paesi, ma tutti i paesi, è di 1,8 Mbit / s.
- In media, questo è un dato davvero molto buono, ma come possiamo vedere dal grafico, gli estremi sono molto ampi.



# Architettura del Computer

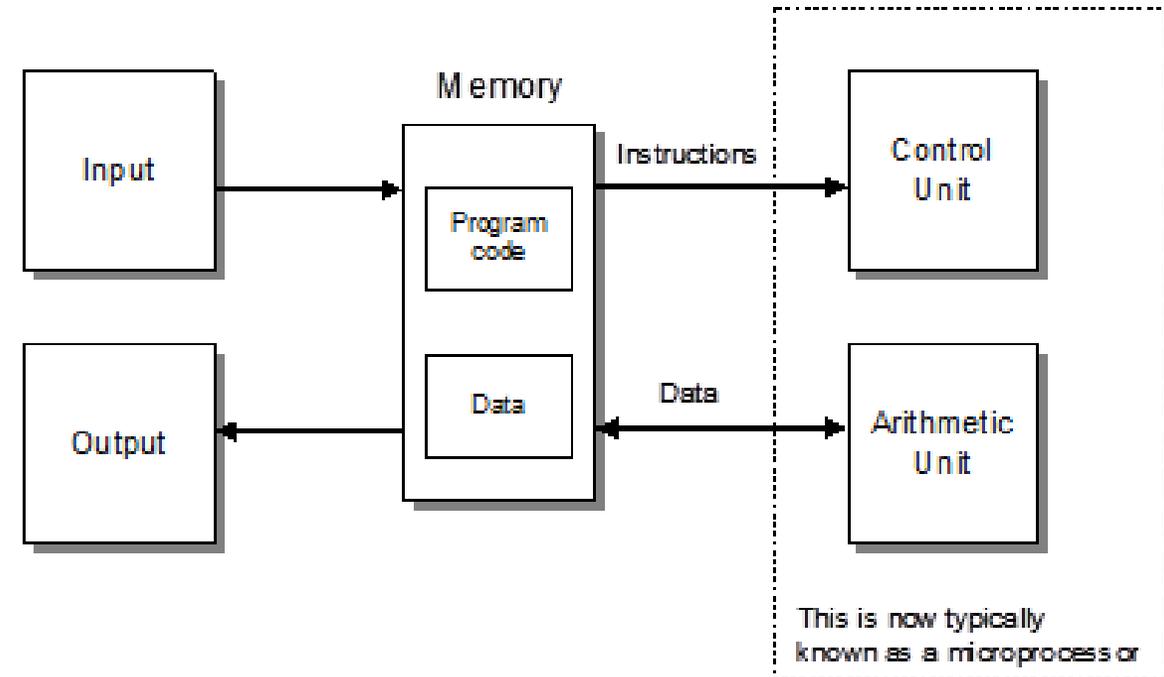
Vediamo ora quali sono i «blocchi» fondamentali che costituiscono il computer, sia esso un «Personal Computer» un «Main Frame» oppure un «Super Computer».

Ogni computer del mondo, indipendentemente dalle sue prestazioni o dal suo costo, può essere ricondotto ad un modello unitario: uno schema di «blocchi funzionali» che lo descrive in termini generici.

## **ARCHITETTURA DI VON NEWMANN**

# Architettura del Computer

Il modello di von Neumann è di notevole importanza in quanto descrive l'architettura hardware su cui è basata la maggior parte dei moderni computer programmabili. Fu sviluppato negli anni '40 dal matematico ungherese John von Neumann nell'ambito del progetto EDVAC\*.



\*Von Neumann non fu l'unico progettista di EDVAC, ma fu lui a descriverne l'architettura in un articolo del 1945.

# Architettura del Computer – I/O

Da una parte abbiamo l'uomo, dall'altra abbiamo il computer. Due linguaggi completamente diversi. Serve qualcosa che consenta di trasferire l'informazione da una parte all'altra. Questa prima componente è una

**«unità di ingresso al mondo dei BIT»**

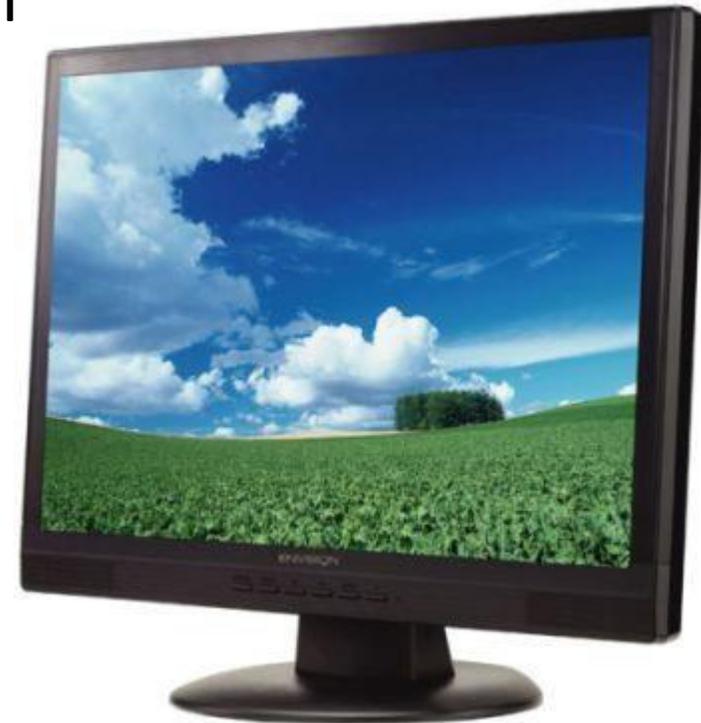
È un mediatore linguistico capace di acquisire informazioni dall'Utente ovvero dal mondo esterno e di trasferirle al Computer.



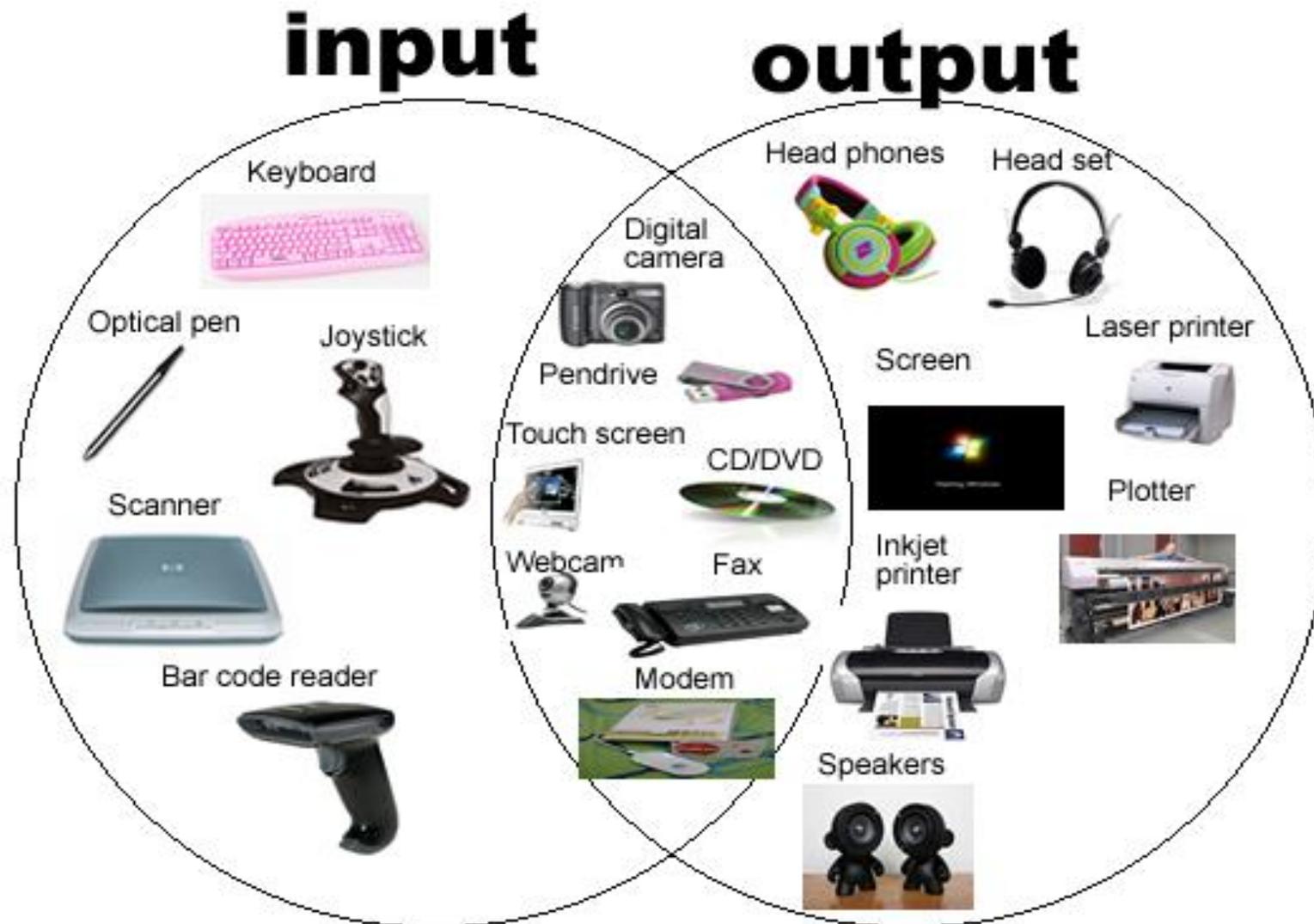
# Architettura del Computer – I/O

In modo del tutto simmetrico abbiamo necessità di trasferire informazione dal mondo del Computer al mondo Esterno: al mondo degli Esseri Umani. Dati che sono codificati in forma di BIT all'interno della memoria del computer, debbono essere trasferiti in forma che sia per noi comprensibile.

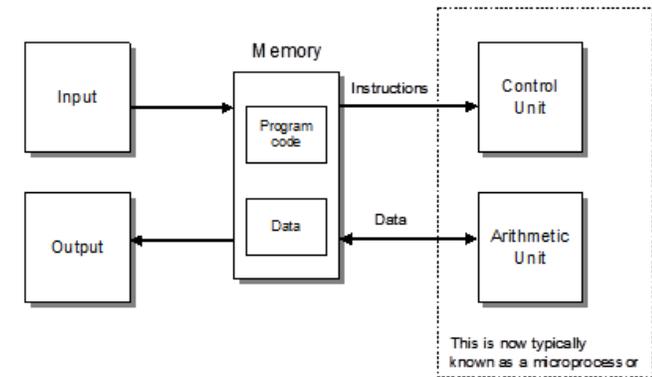
**«unità di uscita dal mondo dei BIT»**



# Architettura del Computer – I/O



# Architettura del Computer – CPU



La **Central Processing Unit (CPU)** o Unità di Elaborazione in italiano risiede all'interno del computer. È il suo cuore e ha la funzione di:

**«elaborare i dati ricevuti in ingresso, secondo gli ordini impartiti dal programma»**

Ma questo può essere fatto solo con l'ausilio di un'altra unità, ossia la

**«Unità di Memoria»**

nella quale risiedono sia i dati da elaborare, sia i programmi da eseguire.

# Architettura del Computer – Memoria

L'Unità di Memoria, come detto, ha la funzione di contenere sia i dati da elaborare, sia il programma delle istruzioni da eseguire.

In prima approssimazione possiamo distinguere due tipi fondamentali di memoria:

- La Memoria centrale RAM (Random Access Memory);
- La Memoria di Massa (Hard Disk).



La prima è una memoria «volatile» che ha la funzione di supportare le attività di elaborazione (alta velocità, bassa capacità);

La seconda è «permanente» e serve a memorizzare i dati e i programmi che debbono poter essere utilizzati anche in più sessioni di utilizzazione del computer.

# Prossimo Capitolo – COME FUNZIONA IL COMPUTER

- Nel prossimo capitolo vedremo con maggior dettaglio come è fatto un computer sia dal punto di vista dell'Hardware, sia dal punto di vista del Software:
  - Come funziona la CPU
  - Cosa è un Sistema Operativo
  - Cosa è un Programma Applicativo