



Tecnologie Informatiche MsOffice

anteprima 50 pagine

## Il Progetto Book in Progress

La rete Book in Progress è nata per produrre libri di testo, di elevato spessore scientifico e comunicativo, scritti dagli stessi docenti, appartenenti a più di cento scuole del secondo ciclo e del primo ciclo della rete nazionale, con istituto capofila l'ITIS "E. Majorana" di Brindisi.

Tale iniziativa migliora significativamente l'apprendimento degli allievi e contemporaneamente fornisce una risposta concreta ai problemi economici delle famiglie e del caro libri.

Con tale iniziativa, si valorizza la funzione docente e si concretizza la personalizzazione degli interventi formativi.

I testi prodotti sono disponibili in formato cartaceo ed in alcuni formati elettronici come pdf, epub e ibooks. Altro elemento positivo, che tale iniziativa porta con sé, è quello di consentire alle famiglie un risparmio di spesa sulla dotazione libraria di circa € 300,00 rispetto ai tetti di spesa previsti dal Ministero.



Nell'edizione 2021 troverete un nuovo capitolo, "Strumenti di produttività online, le google app" realizzato dalla Prof.ssa **Ersilia Pagano**, che illustra alcune web application e i loro enormi vantaggi in termini di sicurezza e collaborazione. Inoltre sono stati introdotti nuovi contenuti aggiuntivi accessibili tramite **QR CODE** "cliccabili", cioè accessibili anche con un "tap" nel caso stiate usando le versioni digitali su dispositivi touch. Abbiamo revisionato e riorganizzato i contenuti relativi al sistema di numerazione nel secondo capitolo. Nel capitolo 6 "Algoritmi e Diagrammi di Flusso", sono stati inseriti nuovi contenuti.

### Si ringraziano gli autori:






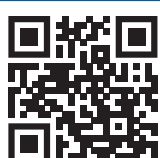



Prof. **Angelo Oliva**, Prof. **Angelo Caputo** – I.I.S.S "Ettore Majorana" – Martina Franca (TA)  
 Prof. **Dario Rinaudo**, Prof. **Domenico Deluso**, Prof.ssa **Graziella Locatelli** – I.S.I.S. "Guglielmo Oberdan" – TREVIGLIO (BG)  
 Prof. **Enrico Sartirana** – I.T.I.S. "Stanislao Cannizzaro" Rho (MI)  
 Prof. **Salvatore Madaro**, Prof.ssa **Giorgia Martina** – I.T.I.S. "E. Fermi" – Francavilla Fontana (BR)  
 Prof. **Paolo Lillo** – I.S.I.S. "L. Scarambone" – Lecce  
 Prof.ssa **Alessandra D'Orazio** – I.S.I.S. "Alessandro Volta" – Frosinone  
 Prof.ssa **Grazia Marzia** – I.I.S.S "C. Mondelli" – Massafra (TA)  
 Prof. **Luca Peresson** dell'I.T.I.S "A. Malignani" – Udine  
 Prof.ssa **Musco Ciriaca** – IISS V. Bachelet" – Gravina in Puglia (BA)  
 Prof.ssa **Ersilia Pagano**, Prof. **Patrizio D'Emilio** dell' ITI "Medi" – S. Giorgio a Cremano (NA)

Rielaborazione unità didattiche, riduzione e impaginazione a cura del Prof. Angelo Oliva dell' I.I.S.S. "Ettore Majorana" – Martina Franca (TA) con la preziosa collaborazione del Prof. Dario Rinaudo dell' I.S.I.S. "Guglielmo Oberdan" - TREVIGLIO (BG)

Maggio 2021

Rete Nazionale Book in Progress  
 Dipartimento Disciplinare d'Informatica  
 Coordinatore Nazionale  
 Prof. Angelo Oliva


**Indice dei contenuti multimediali collegati al testo di tecnologie informatiche**

n	Capitolo pag Descrizione	QR code	Link
1	<b>Capitolo 1 - pag 11</b> Video: L'ENIAC Video d'epoca sul grande calcolatore del 1943 - (You tube)		<a href="https://qrbridge.me/rp1c">https://qrbridge.me/rp1c</a>
2	<b>Capitolo 1 - pag 15</b> Video: Due manager della Silicon Valley ci parlano di CPU, memoria, Input e Output - (You tube) Estratto da Lezione n.4 del percorso "Come funzionano i computer" <a href="https://programmailfuturo.it">https://programmailfuturo.it</a> .		<a href="https://qrbridge.me/s1d8">https://qrbridge.me/s1d8</a>
3	<b>Capitolo 1 - pag 16</b> Video: HW SW, CPU, Alu e Cu, spiegati da due ingegneri di Amazon e MS XBOX - (You tube) Estratto da Lezione n.5 del percorso "Come funzionano i computer" <a href="https://programmailfuturo.it">https://programmailfuturo.it</a> .		<a href="https://qrbridge.me/s1d0">https://qrbridge.me/s1d0</a>
4	<b>Capitolo 1 - pag 17</b> Video: Le componenti del Come è fatto un pc; Treccani Scuola (You tube)		<a href="https://qrbridge.me/s1c7">https://qrbridge.me/s1c7</a>
5	<b>Capitolo 1 - pag 17</b> Video: "una sintesi dei contenuti di questo paragrafo"- Architettura del computer di John Von Neumann, Informaticaso (You tube)		<a href="https://qrbridge.me/s1dp">https://qrbridge.me/s1dp</a>
6	<b>Capitolo 2 - pag 21</b> Video: codifiche dei dati, numeri, video, immagini e suoni - (You tube) Dati e Sistema Binario Lezione n.2 del percorso "Come funzionano i computer" del progetto <a href="https://programmailfuturo.it">https://programmailfuturo.it</a>		<a href="https://qrbridge.me/t2lr">https://qrbridge.me/t2lr</a>
7	<b>Capitolo 2 - pag 23</b> Video. conversione da base 10 a base b - (You tube); Video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="https://qrbridge.me/t2k3">https://qrbridge.me/t2k3</a>
8	<b>Capitolo 2 - pag 25</b> Video. conversione da base b a base 10 - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="https://qrbridge.me/t2l3">https://qrbridge.me/t2l3</a>
9	<b>Capitolo 2 - pag 26</b> Video. Conversioni dirette 2 - 8 - 16, con tabelle di conversione - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="https://qrbridge.me/t2l9">https://qrbridge.me/t2l9</a>
10	<b>Capitolo 3 - pag 53</b> Video. And e OR - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/9n80">http://qrbridge.me/9n80</a>

n	Capitolo pag Descrizione	QR code	Link
11	<b>Capitolo 3 - pag 54</b> Esercizi on line: Esercizi sui primi concetti delle Algebre di Boole - (BookWidget) esercizio autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="https://www.bookwidgets.com/play/Xp7ps1pT-iQAFNH4sfAAAA/GCH24G2/le-proposizioni">https://www.bookwidgets.com/play/Xp7ps1pT-iQAFNH4sfAAAA/GCH24G2/le-proposizioni</a>
12	<b>Capitolo 4 - pag 61</b> Video: "tipologie, funzioni e struttura del sistema operativo" - Architettura del computer di John Von Neumann; Informaticaso (You tube)		<a href="https://qrbridge.me/s1e1">https://qrbridge.me/s1e1</a>
13	<b>Capitolo 5 - pag 73</b> Video: "Dal POWER ON alla visualizzazione del desktop - cosa succede nel computer" - Funzionamento interno di un calcolatore in 3D; Marco De Crescenzo e Stefano Di Silvio (You tube)		<a href="https://qrbridge.me/rozq">https://qrbridge.me/rozq</a>
14	<b>Capitolo 5 - pag 75</b> Video: "La fase di Bootstrap" - Bootstrap di un computer; Informaticaso (You tube)		<a href="https://qrbridge.me/s1e9">https://qrbridge.me/s1e9</a>
15	<b>Capitolo 5 - pag 76</b> Video: "Programmi e Processi in esecuzione"; Informaticaso (You tube)		<a href="https://qrbridge.me/s1ey">https://qrbridge.me/s1ey</a>
16	<b>Capitolo 6 - pag 78</b> Video: Ted Talk: Garry Kasparov parla della competizione Uomo Macchina e dell'Intelligenza Artificiale - TED - (You tube)		<a href="https://qrbridge.me/tey1">https://qrbridge.me/tey1</a>
17	<b>Capitolo 6 - pag 83</b> Video: Algoritmizzazione algoritmi - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/9n7b">http://qrbridge.me/9n7b</a>
18	<b>Capitolo 6 - pag 83</b> Video: Flowgorithm, Software di progettazione di Diagrammi di Flusso - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/9n7y">http://qrbridge.me/9n7y</a>
19	<b>Capitolo 11- pag 278</b> Video: MCD con algoritmo di Euclide in Python - (You tube) video autoprodotta dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/9n83">http://qrbridge.me/9n83</a>
20	<b>Capitolo 12 - pag 293</b> App: TRIS con App Inventor - (You tube) App realizzata dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/app_tris">http://qrbridge.me/app_tris</a>
21	<b>Capitolo 12 - pag 296</b> App: Calcolatrice con App Inventor - (You tube) App realizzata dal dipartimento d'informatica Book in progress come contenuto aggiuntivo del testo		<a href="http://qrbridge.me/bjny">http://qrbridge.me/bjny</a>



**INDICE**

<b>1. CONCETTI DI BASE DELL'I.C.T .....</b>	<b>9</b>	<b>4. IL SISTEMA OPERATIVO.....</b>	<b>61</b>
(autori: A. Oliva - Dario Rinaudo)		(autore: L. Peresson rielaborazione a cura di D. Rinaudo e A. Oliva)	
- Introduzione all'informazione e alla comunicazione ..	9	- I diversi sistemi operativi.....	61
- T.I.C. e competenze digitali .....	10	- La struttura di un sistema operativo .....	62
- Un po' di storia: dai grandi elaboratori al personal computer.....	11	- L'utilizzo dell'interfaccia grafica .....	66
· Il modello di Von Neumann .....	12	- Le principali operazioni del S.O. mediante l'interfaccia grafica.....	68
· il contributo italiano all'informatica.....	12	· Desktop, Barra delle applicazioni, Caratteristiche tecniche del computer.....	68
- Nozioni fondamentali dell'Informatica .....	13	· Ridurre a icona, Ridimensionare.....	68
· Hardware, Software, Copyright.....	13	· La memoria di massa, Backup, Cestino .....	70
· Le più diffuse tipologie di computer.....	14	· Funzione "cerca", Compressione, Virus informatici, Gestione delle stampanti .....	71
· Componenti del personal computer.....	15	· Pannello di controllo .....	71
· Scheda Madre, Cpu, memorie.....	15	· Gestire gli account, Rimuovere dei programmi....	72
· Porte e dispositivi di Input e Output .....	17		
- Virus e antivirus .....	18		
- Salute, sicurezza e ambiente.....	18		
<b>2. CODIFICA DELL'INFORMAZIONE .....</b>	<b>19</b>	<b>5. COME FUNZIONA UN ELABORATORE .....</b>	<b>73</b>
(autori: A. D'Orazio – G. Martina – P. Lillo rielaborazione a cura di D. Rinaudo e A. Oliva)		(autore: A. Oliva)	
- La codifica dell'informazione: il bit .....	19	- Introduzione .....	73
- Codifica dei numeri .....	21	- Accensione del computer .....	73
- Approfondimento:Operazioni con i numeri binari ...	27	· Power On.....	73
- Codifica dei caratteri .....	33	· Si attiva l'alimentatore .....	74
- Codifica delle immagini (fisse).....	35	· Scheda Madre e CPU on .....	74
- Codifica dell'audio.....	41	· Fase di POST.....	74
- Codifica del video.....	42	· Fase di Bootstrap.....	75
- Classificazione dell'informazione .....	42	- Programmi e processi .....	76
- Laboratorio: Conversioni con OpenOffice Calc .....	43		
<b>3. CONNETTIVI LOGICI DELL'ALGEBRA DI BOOLE....</b>	<b>49</b>	<b>6. ALGORITMI E DIAGRAMMI DI FLUSSO .....</b>	<b>77</b>
(autore: Angelo Oliva)		(autore: A. Oliva)	
- Introduzione .....	49	- Introduzione .....	77
- Le Proposizioni e i predicati.....	50	- Problemi e soluzioni .....	78
- Composizione di proposizioni semplici.....	50	- Definizioni e proprietà degli algoritmi .....	79
- L' AND, la congiunzione logica .....	51	- Linguaggi di descrizione degli algoritmi .....	81
- L' OR, la disgiunzione logica .....	52	- I diagrammi di flusso .....	81
- Il NOT, la negazione logica .....	53	- Variabili e Istruzioni .....	82
- Symbolismi e significati .....	54	- Strutture fondamentali e la programmazione strutturata.....	83
- Espressioni Booleane.....	54	- La sequenza .....	84
- Ordini di priorità dei connettivi logici .....	56	- La selezione .....	84
- Gli operatori logici nei fogli elettronici.....	56	- L'iterazione o ciclo .....	85
- Approfondimento .....	60	- Esempi fondamentali:	
		Lo scambio e la variabile temporanea .....	90
		Esempi fondamentali: La variabile accumulatore.....	92
		Esempi fondamentali: La variabile contatore .....	93
		Debug e correzione di errori:	
		la tecnica della trace table .....	94
		Conclusioni .....	95

**LABORATORIO****7. STRUMENTI DI PRODUTTIVITÀ ONLINE,****LE GOOGLE APPS .....97**

(autore: Ersilia Pagano Revisione: Angelo Oliva)

- Dal cloud al cooperative learning ..... 97
- Google Apps ..... 98
- Facciamo una breve panoramica: ..... 99
- Documenti Google ..... 99
- Creare un nuovo Documento google ..... 100
- Modificare e formattare un documento google .... 101
- Condividere un documento google ..... 102
- La barra dei menù .....103
- Esplora ..... 105
- Commenti e Suggestimenti ..... 109
- Organizzare dinamicamente la struttura del documento con gli stili paragrafo ..... 111
- Con Documenti Google converti un'immagine in testo .....111
- Scaricare il Documento Google in altri formati .... 112
- Cronologia .....112

**8. ELABORAZIONE TESTI (Microsoft Word) .....115**

(autore: Dario Rinaudo)

- L'interfaccia di Word ..... 115
- Indicazioni di carattere generale ..... 116
- Come salvare un documento ..... 116
- Come aprire un documento ..... 116
- Come uscire dal programma ..... 117
- La guida di Word ..... 117
- Le funzioni dei pulsanti ..... 117
- Anteprima di stampa e la stampa ..... 118
- Scelta carattere e dimensione ..... 118
- L'allineamento del testo ..... 119
- Il grassetto, il corsivo e la sottolineatura ..... 119
- I comandi annulla e ripristina ..... 121
- La visualizzazione dei segni di paragrafo ..... 121
- Come selezionare il testo o una parte di testo ... 122
- Interlinea ..... 124
- Come cambiare colore al carattere ..... 125
- Il menu di scelta rapida ..... 126
- L'evidenziatore ..... 126
- Alcune scorciatoie da tastiera ..... 127
- Gli elenchi puntati, numerati e alfabetici ..... 128
- Come correggere un testo ..... 130
- La sillabazione ..... 132
- I bordi e lo sfondo ..... 133
- Le caselle di testo ..... 135
- WordArt ..... 136
- Come inserire immagini clipart ..... 139
- Come inserire immagini nel foglio ..... 141
- Il capolettera ..... 143

- I comandi trova, sostituisci e vai ..... 145
- Come inserire simboli e caratteri speciali ..... 146
- Come applicare la rientranza nella prima riga .... 148
- Alcune note sulla formattazione del carattere .... 149
- Studio dei comandi taglia, copia, incolla e copia formato ..... 150
- Come inserire forme ..... 152
- Impostazioni di pagina ..... 154
- Come inserire il numero di pagina ..... 156
- Come inserire note a piè di pagina ..... 157
- Come disporre un testo su più colonne ..... 160
- Generalità intestazione e piè di pagina ..... 161
- Le tabulazioni ..... 163
- Le tabelle ..... 165
- Come ordinare i dati nella tabella ..... 167

**9. IL FOGLIO ELETTRONICO (Microsoft Excel) .....175**

(autore: Dario Rinaudo)

- Come salvare un lavoro, come aprirlo ..... 176
- Indicazioni operative ..... 176
- Come modificare la larghezza delle colonne ..... 177
- La selezione ..... 178
- Come inserire righe e colonne..... 179
- Come inserire WordArt..... 179
- Come spostarsi da un foglio di lavoro ad un altro. 179
- Come effettuare calcoli..... 180
- I pulsanti maggiormente usati ..... 182
- I bordi e le linee ..... 183
- La stampa, i margini, intestazione a piè di pagina 184
- La copia delle formule ..... 188
- Il quadratino di riempimento..... 190
- I decimali ..... 191
- Il pulsante unisci e centra ..... 192
- La disposizione del testo in verticale ..... 194
- L'uso delle funzioni ..... 195
- La media ..... 197
- Il valore minimo e massimo ..... 197
- Generalità sui riferimenti relativi e assoluti ..... 199
- Come rinominare un foglio di lavoro ..... 201
- Costo medio ponderato ..... 201
- I calcoli percentuali..... 202
- La protezione del foglio di lavoro ..... 204
- Concetti generali sulla fattura ..... 204
- Generali sullo scorporo dell'IVA ..... 205
- Creazione di un grafico ..... 207
- Alcuni messaggi di errore ..... 212

## 10. STRUMENTI DI PRESENTAZIONE

### (Microsoft PowerPoint).....213

(autore: Graziella Locatelli)

- Introduzione ..... 213
- Apertura ..... 213
- Finestra di PowerPoint ..... 214
- Le schede della Barra Multifunzione ..... 214
- Uscire da PowerPoint ..... 217
- Nominare e salvare una presentazione ..... 217
- Riaprire una presentazione ..... 217
- Realizzare una presentazione ..... 218
- Creare una presentazione basata su un Tema predefinito ..... 219
- Animare una presentazione ..... 225
- Avviare una presentazione ..... 226
- Grafici e diagrammi ..... 227
- Creare una presentazione con collegamenti ipertestuali ..... 228
- Creare un collegamento ipertestuale ..... 232
- Aggiungere un suono ..... 233
- Stampare le diapositive ..... 233
- Esportare diapositive in Word ..... 234
- Importare slide da altre presentazioni ..... 234
- Esercizi:
  - Presentazione “La Libertà” ..... 235
  - Presentazione “I diagrammi di Flusso” ..... 237
  - Presentazione “Le sette meraviglie del mondo moderno” ..... 238

## 11. CODING: dal Problema al Programma con Scratch e Python .....239

(Autori: Domenico Deluso, Enrico Sartirana, Rielaborazione a Cura di A. Oliva)

- Introduzione: Pensiero computazionale e coding ..239
- L'ambiente Scratch (Autore: Domenico Deluso).....240
  - Stage ..... 240
  - Script ..... 241
  - Inizio/Fine ..... 242
- Algoritmi ..... 243
  - Input e Output..... 243
  - I costrutti ..... 243
  - Sequenza ..... 243
  - Selezione ..... 243
  - Iterazione ..... 244
  - Condizioni ..... 244
  - Assegnamento ..... 244
  - Dall'Algoritmo a Scratch ..... 245
  - Un esempio ..... 245
  - Schema di lavoro ..... 249

- La sequenza ..... 249
  - Esempio 1 ..... 249
  - Esempio 2 ..... 251
  - Esempio 3 ..... 253
- La Selezione ..... 255
  - Esempio 1 ..... 255
  - Esempio 2 ..... 256
  - Esempio 3 ..... 257
- L'Iterazione ..... 259
  - Esempio 1 ..... 259
  - Esempio 2 ..... 260
  - Esempio 3 ..... 262
  - Esempio 4 ..... 265
- Il linguaggio Python (Autore: Enrico Sartirana) .....267
- Valori, tipi, variabili, operazioni .....268
- Input, Elaborazione, Output .....271
- Operare una scelta: la selezione.....274
- Ripetere più volte le istruzioni: i cicli, o loop.....276
- Conclusioni .....278
- Esercizi.....279

## 12. CODING CON APP INVENTOR .....281

(Autori: G. Marzia – A. Caputo)

- Creare app con “App Inventor” ..... 281
- Esercitazione 1 l'app: “Parlami” ..... 283
- Esercitazione 2 l'app: “Scrivi e ascolta” .....287
- Esercitazione 3 l'app: “Scegli il colore e disegna” ..290
- Esercitazione 4 l'app: “Il gioco del tris” .....292
- Esercitazione 5 l'app: “La calcolatrice” .....294
- Tabella dei Componenti User Interface .....297

NOTE

anteprima 50 pagine

# 1. Concetti di base dell'ICT



Autore: Angelo Oliva

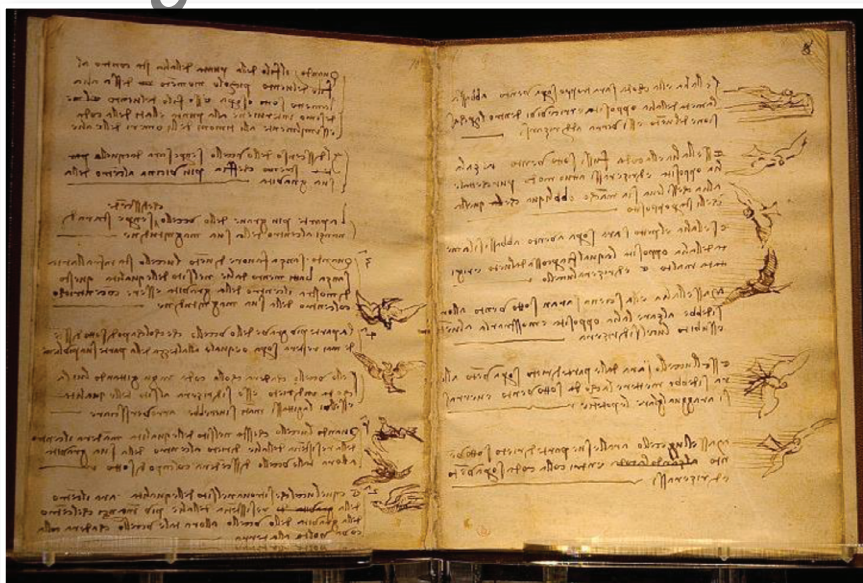
I contenuti delle nozioni fondamentali dell'informatica sono rielaborati dall'unità didattica a cura di **DARIO RINAUDO**

Competenze	Abilità	Conoscenze
Utilizzare linguaggio tecnico in merito a componenti e problematiche di funzionamento del personal computer e in generale sul mondo delle tecnologie informatiche	Riconoscere le componenti interne ed esterne del personal computer	Nozioni storiche, generali e fondamentali della Tecnologia dell'informazione e della comunicazione



## Introduzione all'informazione e alla comunicazione

Quante volte sarà stato scoperto il fuoco prima che questo sia stato sistematicamente utilizzato dall'uomo? Chi sono stati i primi uomini a usare la scrittura? l'inchiostro? la polvere da sparo? Probabilmente in più luoghi della terra sono state fatte le stesse scoperte e le stesse invenzioni, senza che gli autori fossero a conoscenza delle esperienze già fatte da altri. Le scoperte scientifiche e le invenzioni erano destinate a scomparire con i loro inventori se queste non fossero comunicate in qualche modo all'umanità, ecco perché la comunicazione è stata un processo fondamentale per il progresso. *La comunicazione è un processo di trasmissione d'informazioni che prevede l'esistenza di una sorgente emittente e di un ricevente o destinatario e infine l'utilizzo di un canale, cioè di un mezzo di trasmissione.* Il tipo di canale utilizzato è fondamentale sull'esito della comunicazione: se si usa la voce e l'etere si ha sicuramente un numero limitato di destinatari, mentre un messaggio stampato o trasmesso via radio avrà potenzialmente un numero maggiore di destinatari. Non sempre il solo mezzo garantisce la diffusione dell'informazione, basti pensare ai progetti delle macchine di Leonardo da Vinci, di cui l'inventore ha lasciato accurata documentazione su carta, ma ciò nonostante, tali progetti sono rimasti sconosciuti per secoli a causa anche della modalità che Leonardo utilizzava nella scrittura, detta speculare perché consiste nello scrivere le lettere come se fossero riflesse da uno specchio. Con l'invenzione della stampa, si diffusero libri e pubblicazioni scientifiche, che consentirono agli scienziati di condividere le informazioni sui propri esperimenti, iniziò così quel processo che portò sempre più rapidamente alle attuali tecnologie.



L. Da Vinci codice sul volo degli uccelli – Fotografo Luc Viatour - [www.lucnix.be](http://www.lucnix.be) - CC BY-SA 3.0

## T.I.C. e competenze digitali

Sebbene sia nato per eseguire calcoli complessi, il computer oggi è diventato uno strumento rapido e flessibile per rappresentare, gestire e comunicare informazioni. Le molteplici attività legate all'uso delle più recenti apparecchiature informatiche sono identificate col termine **TIC**, acronimo ottenuto da *Tecnologie dell'Informazione e della Comunicazione*, che poi deriva dalla traduzione di *Information and Communication Technology* e dal corrispondente acronimo **ICT**.

La diffusione della rete internet e di dispositivi come smartphone, tablet e Personal Computer in tutte le case, consente, e a volte rende obbligatorio, il loro utilizzo non solo per svago ma anche per gestire molteplici attività come la gestione del conto corrente bancario o il rapporto con la Pubblica Amministrazione, al punto che si parla oggi di **Home Office** ("ufficio casalingo"). Per questo si rende necessario per il cittadino di oggi possedere tra le altre competenze chiave anche le competenze digitali.

La raccomandazione del Parlamento e del Consiglio europei del 18 dicembre 2006 dell'Unione Europea definisce le otto competenze chiave per l'apprendimento permanente "Key Competences for Lifelong Learning" e tra queste la quarta è la Competenza digitale che:

«Consiste nel saper utilizzare con dimestichezza e spirito critico le tecnologie della società dell'informazione per il lavoro, il tempo libero e la comunicazione.

Essa è supportata da abilità di base nelle Tic: l'uso del computer per reperire, valutare, conservare, produrre, presentare e scambiare informazioni, nonché per comunicare e partecipare a reti collaborative tramite Internet».

Il cittadino che si avvale di questi servizi elettronici, oggi detto **e-citizen**, troverà sempre più spesso dei vantaggi anche sotto l'aspetto economico.

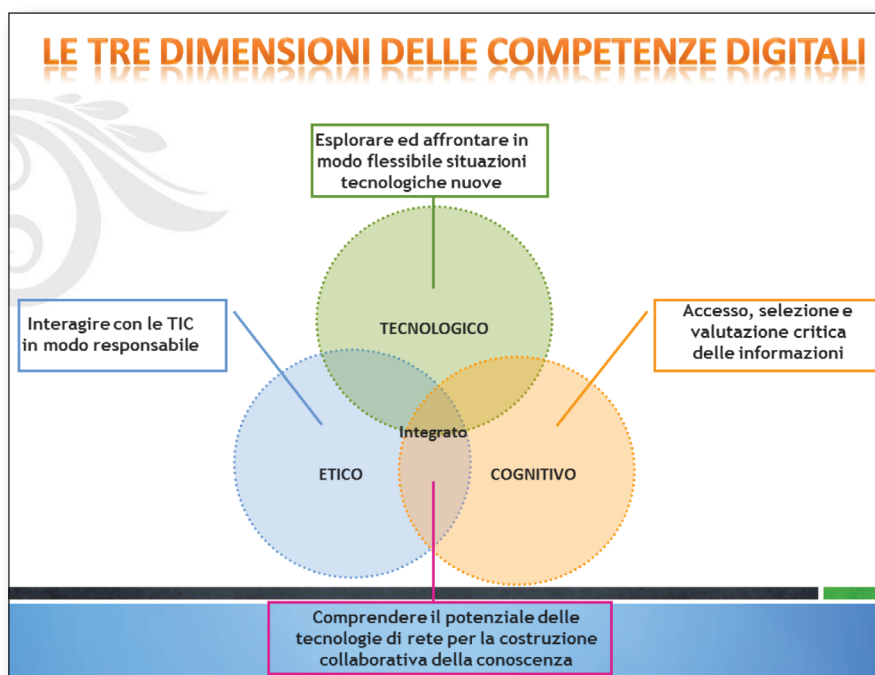
Gli studi in corso hanno portato alla strutturazione della competenza digitale in tre dimensioni:

- **dimensione tecnologica:**  
riconoscere le criticità tecnologiche e le interfacce, selezionare la tecnologia adeguata per ciascun compito, operare logicamente, rappresentare processi simbolici, distinguere tra reale e virtuale.
- **dimensione cognitiva:**  
saper trattare ovvero sintetizzare, rappresentare e analizzare testi, dati, tabelle e grafici e saper valutare la pertinenza dell'informazione e la sua affidabilità.
- **dimensione etica:**  
conoscere i concetti di tutela della privacy, rispettare i diritti intellettuali dei materiali reperiti in Internet e l'immagine degli altri, comprendere il dislivello sociale e tecnologico che può esistere tra paesi, persone, generazioni, e il problema dell'accessibilità.

### "Key Competences for Lifelong Learning"

- 1) Communication in the mother tongue;
- 2) Communication in foreign languages;
- 3) Mathematical competence and basic competences in science and technology;
- 4) **Digital competence;**
- 5) Learning to learn;
- 6) Social and civic competences;
- 7) Sense of initiative and entrepreneurship;
- 8) Cultural awareness and expression.

[http://ec.europa.eu/education/policies/2010/doc/keyrec\\_it.pdf](http://ec.europa.eu/education/policies/2010/doc/keyrec_it.pdf)

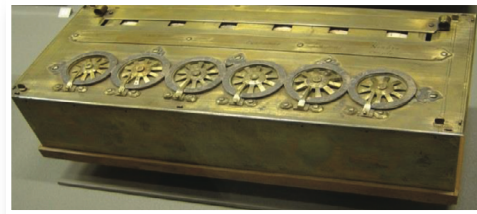




## Un po' di storia: dai grandi elaboratori al personal computer.

Il termine "Computer deriva dal latino "computo" (contare). In effetti, si tratta di una macchina specializzata per eseguire calcoli. Ispirandosi all'**abaco** **Blaise Pascal** nel 1642, inventò la pascalina, uno strumento meccanico in grado di eseguire addizioni con riporto.

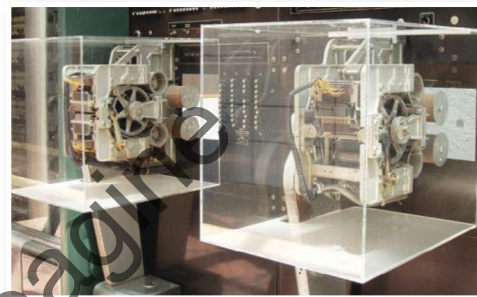
**Charles Babbage** realizzò nel 1850 una macchina capace di ricevere una serie di istruzioni che avrebbe potuto svolgere anche non immediatamente ma successivamente. Fu questo lo spunto per realizzare macchine adibite all'esecuzione di operazioni ripetitive con l'utilizzo di schede, che trovarono grande applicazione nel campo dell'industria tessile.



La Pascaline, esposta al Musée des Arts et Métiers, di Parigi, Fotografia di David Monniaux

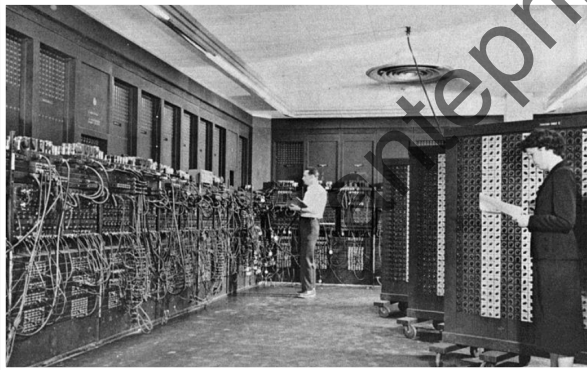
Durante la seconda guerra mondiale lo sviluppo si intensificò, grazie alle intuizioni di scienziati come **Turing** e **Von Neumann**, arrivando alla realizzazione, nell'immediato dopoguerra, dei primi elaboratori il **Mark1** e l'**ENIAC**.

La costruzione del **Mark1** iniziò nel 1939 al dipartimento di fisica dell'Università di Harvard (Cambridge, Massachusetts, U.S.A.) e terminò nel 1943 presso i laboratori dell'IBM ad ENDICOTT (New York), che finanziò il progetto per un importo finale di 250.000 dollari. Esso era costituito da 765.000 componenti, interruttori, relè, alberi di rotazione e frizioni e centinaia di chilometri di cavi, occupava una lunghezza di 16 m, si sviluppava in altezza per 2,4 m ed aveva una profondità di circa 0,5 metri; pesava circa 4 tonnellate e mezzo. Il Mark I poteva memorizzare 72 numeri di 23 cifre decimali ciascuno.



Mark I - unità Input-Output CC BY-SA 3.0  
Caricato da Daderot su en.wikipedia

Poteva eseguire tre addizioni o sottrazioni al secondo, una moltiplicazione in 6 secondi, una divisione in 15,3 secondi ed un logaritmo oppure una funzione trigonometrica in più di un minuto. Il Mark I leggeva le sue istruzioni su delle schede perforate e, eseguita l'istruzione corrente, passava alla successiva. I programmi complessi erano fisicamente lunghi.



L'ENIAC al Ballistic Research Laboratory Philadelphia, Pennsylvania

L'**ENIAC** fu progettato e costruito alla Moore School of Electrical Engineering un'ex scuola universitaria dell'Università della Pennsylvania, allo scopo di realizzare di una macchina da calcolo capace di risolvere i problemi di calcolo balistico per il lancio dei proiettili di artiglieria. Per la sua realizzazione, terminata nel 1946 con una spesa complessiva otto volte maggiore di quella preventivata e pari a circa 486.800 dollari, furono necessarie ben 18.000 valvole termoioniche che portarono l'ambiente ad una temperatura superiore ai 50 °C. Purtroppo, infatti, il transistor che soppianderà la valvola termoionica nella realizzazione del computer elettronico fu inventato solo un anno. L'ENIAC occupava una superficie di 167 mq e pesava oltre 30 tonnellate, pur eseguendo calcoli elementari e assorbiva così tanta energia

elettrica che, alla sua prima messa in funzione, causò un black-out nel quartiere ovest di Filadelfia. John Von Neumann lo utilizzò per eseguire la prima previsione meteorologica al computer. L'ENIAC rimase in funzione fino al 2 ottobre 1955, fu poi trasferito a Washington, al museo Smithsonian Institution, dove è ancora esposto.



John Von Neumann - Budapest  
28/12/1903 Washington 8/02/1957



L'ENIAC Video  
d'epoca sul grande  
calcolatore del 1943

Lo scienziato **John Von Neumann** nel 1946 pubblicò un articolo sul progetto di una macchina per il calcolo automatico che introduceva il concetto di programma registrato in memoria. Il calcolatore digitale moderno nasce all'inizio degli anni '40 per merito di John von Neumann, J.Presper Eckert e John William Mauchly. Questo team di scienziati concettualizzò l'idea di macchina con accesso casuale e non sequenziale ai dati in memoria.



(RAM Random Access Machine). Il calcolatore ha ancora una memoria organizzata come sequenza di celle d'informazione, ma il tempo di accesso a una cella non dipende più dalla posizione della cella stessa, in pratica non si devono scorrere le celle sino a trovare quella cercata, ma si accede a essa direttamente.

Tale macchina, nota come **Modello di Von Neumann** è stata ed è tuttora l'architettura su cui si basano i moderni calcolatori. La sua caratteristica fondamentale che la memoria contiene sia i **programmi** sia i **dati**.

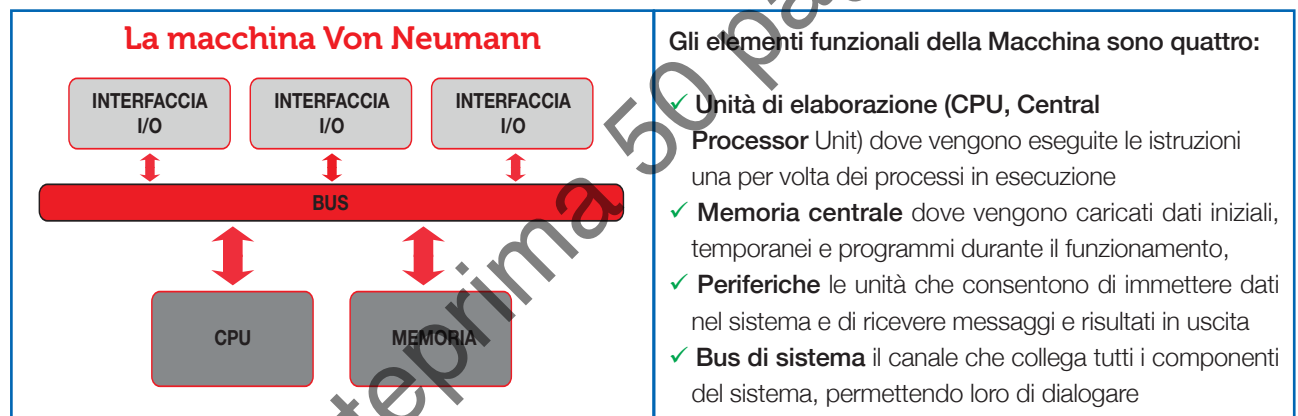
Il programma è:

- ✓ la sequenza d'istruzioni che devono essere eseguite
- ✓ i dati sono le informazioni che il programma elabora per produrre il risultato.

Queste informazioni includono:

- ✓ i dati d'ingresso
- ✓ i dati di lavoro, cioè quei dati parziali ottenuti durante l'elaborazione
- ✓ i risultati della computazione.

Questa copresenza di dati e programmi nella stessa memoria di lavoro rappresenta la fondamentale differenza tra le macchine costruite o pensate sino a quel momento e i moderni calcolatori digitali. Da questo momento, infatti, è possibile che un programma possa scrivere nell'area di memoria di un altro un altro programma, consentendo una nuova serie di tecniche di programmazione importantissime, come quella detta del "Produttore Consumatore" dove appunto una procedura o programma può generare dati che vengono utilizzati da un altro programma anche ciclicamente.



Nel 1962, Philippe Dreyfus conia il termine **informatica**, dalle parole *information electronique ou automatique*, per definire il trattamento automatico dell'informazione mediante calcolatore.

L'uso dei transistor e la miniaturizzazione consentirono la realizzazione, già negli anni '70, dei primi **Home Computer**, computer casalinghi per applicazioni molto elementari, la gestione di piccoli archivi e principalmente per videogiochi, molto noti all'epoca erano i Commodore VIC 20 e il Commodore CBM 64, dove 64 indicava i Kilobyte di Ram, il Sinclair Spectrum, e l'Amiga di Atari. Nel 1981 arrivano sul mercato i primi **Personal Computer IBM**, che iniziarono a soddisfare le esigenze di piccole aziende nelle classiche operazioni di corrispondenza, calcolo, disegno tecnico, fino alla gestione dell'intera azienda.

Non si può dimenticare il **contributo italiano all'informatica**, la Società Olivetti sotto la guida di Adriano Olivetti nel 1940 produsse la sua prima macchina addizionale Olivetti, seguita nel 1945 dalla Divisumma 14, la prima calcolatrice scrivente al mondo in grado di eseguire le quattro operazioni, queste erano macchine completamente meccaniche. Nel 1959 Olivetti sviluppa



Un esemplare della Programma 101 esposta al Museo della storia del Computer di Mountain View – Fotografo Bill Abbott: Olivetti printing calculator/computer su Flickr CC BY-SA 2.0

uno dei primi mainframe computer a transistor: **Elea 9003**. Il prodotto però più rivoluzionario di Olivetti risulterà, la **Programma 101** ritenuta oggi il primo personal computer, progettato dall'ing. **Pier Giorgio Perotto** e presentato insieme alla Logos 27, un'efficientissima calcolatrice meccanica, alla fiera di New York del 1965. Probabilmente neanche l'azienda poteva prevedere il potenziale e il successo di tale prodotto tanto che pur disponendo di un grandioso stand allestito per la Logos, la P101 venne relegato in una saletta di fondo, dopo che i primi visitatori si accorsero delle caratteristiche della P101, l'afflusso di visitatori fu tale che il personale dello stand dovette improvvisare un servizio d'ordine per l'accesso allo stand. Ciò che colpiva della P101 era la capacità per una macchina che poteva stare sopra una scrivania, da cui il termine desktop computer, di eseguire operazioni, allora piuttosto complesse. La rete televisiva americana NBC acquistò 5 esemplari per computare i risultati elettorali da fornire ai propri telespettatori. Fu l'inizio di un successo mondiale, si ricorda che il deposito del brevetto di alcune soluzioni tecniche della P101 obbligò la Hewlett-Packard meglio nota come HP, a riconoscere un indennizzo di 900.000 dollari all'Olivetti quando le fu contestata la violazione di brevetto. Della P101 furono vendute circa 44.000 unità.

Anche un altro grande personaggio italiano dell'informatica ha iniziato la sua carriera nell'Olivetti, il suo contributo in questo campo è stato così importante che potrebbe essere celebrato al pari di Bill Gates e Steve Jobs, si tratta di **Federico Faggin** nato a Vicenza nel 1941 ed è un imprenditore, inventore e informatico italiano naturalizzato statunitense. Faggin nel 1970 è stato il capo progetto dell'Intel 4004, primo microprocessore al mondo, e dei successori Intel 8008, 4040 e 8080. Fu anche lo sviluppatore della tecnologia MOS con porta di silicio che permise la fabbricazione dei primi microprocessori e delle memorie EEPROM e RAM dinamiche. Nel 1974 Faggin fondò e diresse la ditta Zilog dove dette vita al famoso microprocessore Z80 ancora usato nel 2013. Nel 1986 Faggin fondò e diresse la Synaptics, ditta che sviluppò i primi Touchpad e Touchscreen.



Federico Faggin - Vicenza, 1 dicembre 1941

## Nozioni fondamentali dell'Informatica

### **HARDWARE**

Per **hardware** s'intende tutti ciò che ha una consistenza fisica, un peso, quindi le componenti fisiche del computer compresi i dispositivi elettrici ed elettronici. La tastiera, il mouse, il monitor, le casse acustiche, l'unità centrale, ecc., fanno parte dell'hardware.



### **SOFTWARE**

Per **software** intendiamo tutto ciò che è utile o indispensabile al computer per funzionare e non ha una consistenza fisica, quindi i programmi che servono e permettono al computer di compiere delle azioni. È software il sistema operativo, un videogioco, il programma per elaborare testi, di disegno, i driver di un dispositivo periferico come la stampante ecc.

- Il **software di base** è il sistema operativo, ne sono esempi Windows, l'analogo sistema operativo gratuito e open source Linux, Mac OS della Apple computer, Android, usato su vari dispositivi mobili come tablet e smartphone. Lo scopo del sistema operativo è di riconoscere le risorse del sistema come la quantità di memoria centrale (RAM) e dispositivi di memoria come dischi rigidi o pen-drive, e metterli a disposizione dell'utente attraverso un'interfaccia grafica utente amichevole (GUI Graphic User-friendly Interface)
- Il **software applicativo** è costituito da quella classe di programmi dedicati a svolgere particolari applicazioni, come i software per l'editing video, il fotoritocco oltre che quei software dedicati all'office automation, gli elaboratori di testi, i fogli elettronici o di calcolo, i software per la gestione di database, tali software sono generalmente disponibili in "Suite" di cui le più note sono Microsoft Office, e gli analoghi gratuiti OpenOffice e LibreOffice.

### Classificazione e distribuzione dei software

Un programma venduto e anche quelli distribuiti in varie forme, hanno la licenza d'uso (**EULA - End User Licence Agreement**, cioè l'accordo di contratto del produttore del software con l'utente finale).

Esistono in commercio oltre ai software commerciali, altri tipi di software che in genere, vengono scaricati da internet, esaminiamoli:

- 1) **Open source** (codice aperto): distribuito e copiabile gratuitamente, il cui codice sorgente è accessibile e modificabile, senza violare il diritto d'autore.
- 2) **Freeware** (libero): distribuito e duplicabile gratuitamente, il cui codice non è però accessibile.
- 3) **Shareware** (condiviso): circola liberamente sulla rete e può essere copiato ed utilizzato gratuitamente nei limiti indicati dalla licenza. Possono essere previsti l'utilizzo:
  - a. temporaneo, entro un certo termine;
  - b. parziale, con alcune funzioni disattivate.
- 4) **Public domain** (dominio pubblico): chi lo ha creato non esercita il diritto d'autore, per cui è gratuitamente utilizzabile e duplicabile.

### Ma cos'è il DIRITTO D'AUTORE?

La diffusione del computer e di internet ha reso più pressante la questione della tutela del **copyright** (diritto d'autore). Infatti è noto come sia facile copiare un programma per videogame con il computer o scaricare, da internet, files di qualsiasi genere (immagini, musica,...) e software (programmi per elaboratore). La legislazione italiana sulla protezione del diritto d'autore e di altri diritti connessi al suo esercizio è stata aggiornata per comprendere anche il software e i database, equiparati ad opere dell'ingegno di carattere creativo. Gli autori di opere letterarie, composizioni musicali, disegni, opere cinematografiche e fotografiche erano tutelati nella prima emanazione della legge, oggi colui che crea un software è il titolare dei diritti esclusivi di riprodurre, distribuire, noleggiare o dare in prestito l'opera o esemplari di essa. La SIAE detiene il registro pubblico in cui è trascritto il nome del software e dell'autore che così ne può provare la paternità. Chi abusivamente utilizza, duplica, riproduce le opere munito del contrassegno della SIAE oppure acquista o noleggia attrezzature per eluderne le protezioni è punito con una multa e la confisca del materiale. Oltre la multa, rischia il carcere chi abusivamente duplica, distribuisce, vende e detiene, per trarne profitto, software o banche dati, di cui non è l'autore, oppure ne rimuove le chiavi di protezione.

### Le più diffuse tipologie di computer

Oggi le più diffuse tipologie di computer sono:

**Desktop PC – personal computer:** Il primo PC fu lanciato in commercio dalla IBM nel 1981. E' un elaboratore monoutente cioè pensato per l'uso da parte di una persona per volta. Oggi ha raggiunto prestazioni tali da poter sostenere carichi di lavoro che in passato erano svolti da mainframe e workstation. La velocità di esecuzione già molto elevata, è aumentata ancora con l'introduzione dei processori multicore (dualcore, quadcore), dove il processore, che è l'unità che esegue l'elaborazione delle istruzioni, da l'impressione che le esecuzioni dei vari programmi avvenga contemporaneamente, ma in realtà sono eseguiti ciclicamente uno per volta. Il personal computer è efficacemente utilizzato in luoghi di lavoro o come postazione domestica sia per programmi applicativi professionali, sia ludici, sia per navigare in internet.



**Laptop** – computer portatile, ha raggiunto le stesse prestazioni di un Desktop PC, ma consente la mobilità perchè è fornito di batteria ed ha caratteristiche di dimensioni e peso contenuti. I computer portatili rispetto ai PC, in genere non si prestano a notevoli aggiornamenti hardware.



**Netbook:** è un portatile di dimensioni e prestazioni ridotte, il suo monitor ha la grandezza di pochi pollici ed è usato per semplici applicazioni e per collegarsi ad internet.



Si possono citare anche i seguenti dispositivi che non rientrano nella categoria dei computer, poiché non consentono di fare molte attività, prime fra tutte la programmazione, ma si prestano a svolgere comodamente alcune funzioni come la navigazione in internet ed in particolare nell'utilizzo dei social network e nella gestione della posta elettronica:

**Tablet:** con prestazioni simili a un portatile, ma di dimensione e peso contenuti, in genere non sono dotati di tastiera esterna, l'utente interagisce con tale strumento tramite touch screen. Gli schermi LCD, sono retro-illuminati per cui alla luce del sole, la lettura non è sempre agevole.



Fra gli altri dispositivi portatili bisogna ricordare lo **SMARTPHONE** che è un telefono cellulare e un semplice computer, infatti su esso possono essere installate applicazioni, si possono gestire dati, video, foto e ci si può collegare in rete.



### Componenti Hardware del personal computer

L'Hardware è l'insieme delle componenti fisiche del computer, nella loro descrizione è essenziale una prima classificazione. Iniziamo dal **Case** che è quella scatola generalmente in versione Desktop Tower. Al suo interno sono presenti una serie di dispositivi per il funzionamento del computer. A esso poi si collegano una serie di dispositivi esterni che servono per immettere dati e comandi (dispositivi di input) e per ricevere i risultati delle azioni richieste (dispositivi di output), che sono informazioni di vario formato, testo, immagine, suono, video, e altro, si pensi per esempio all'effetto vibrazione di alcune periferiche di gioco.

I principali dispositivi contenuti all'interno del Case sono:

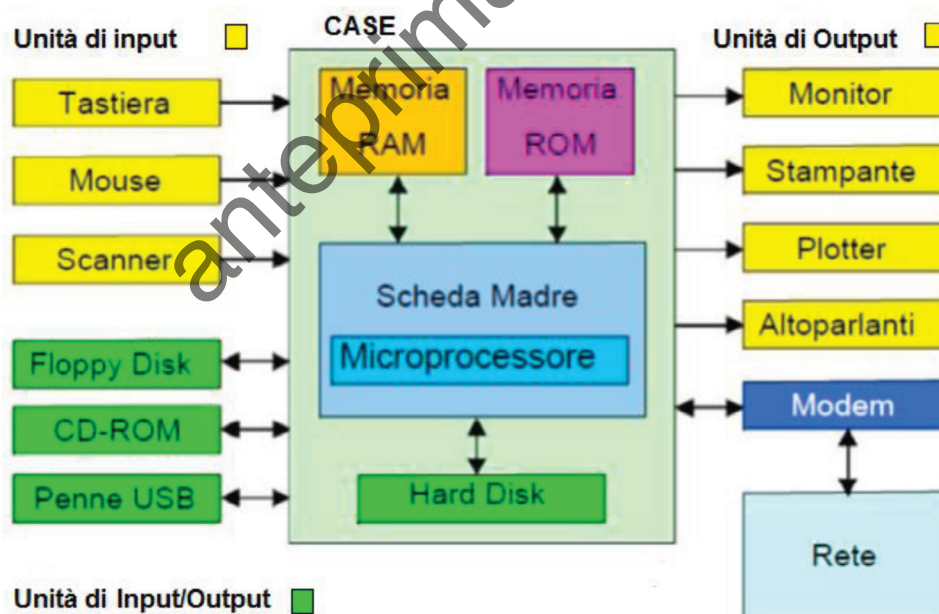
- La Scheda Madre (Mainboard)
- La CPU (Microprocessore)
- Le Porte di comunicazione
- Le Memorie
- Le Memorie di Massa



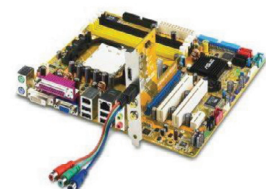
I Principali dispositivi esterni detti Unità Periferiche sono:

- Dispositivi di Input
- Dispositivi di Output
- Dispositivi di Input e Output

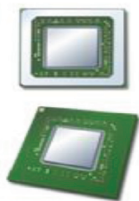
Due manager della Silicon Valley ci parlano di CPU, memoria, Input e Output



La **scheda madre**, spesso chiamata con il suo nome inglese motherboard, è progettata per far dialogare tra di loro le varie componenti del pc, contiene connettori, slot, punti di alimentazione, e un insieme di circuiti integrati, detto chipset. Microprocessori e banchi di memoria Ram e varie schede tra cui quella grafica e quella audio vengono direttamente installate su di essa, così come sul telaio di un'automobile vengono assemblati il motore e le parti meccaniche, elettriche e la carrozzeria.







La componente più importante è La **CPU** (Central Processing Unit) che può essere considerata il cervello del computer, chiamata anche processore o microprocessore. Essa svolge le sue funzioni temporizzando le sue operazioni con l'**impulso di clock**, un segnale sincrono generato da un circuito interno chiamato appunto clock. La velocità di esecuzione delle operazioni della CPU corrisponde alla velocità di emissione di questo impulso, si chiama frequenza di clock e viene misurata in hertz. Se un clock lavora a un GHz (Gigaherz), vuol dire che l'unità centrale esegue in un secondo un miliardo d'istruzioni elementari. Se un clock lavora a un MHz (Megaherz), vuol dire che l'unità centrale esegue in un secondo un milione di istruzioni elementari.

Per eseguire un qualsiasi programma La CPU estrae una per volta le istruzioni ed i dati necessari dalla RAM, con la quale comunica attraverso delle linee di collegamento chiamate **BUS**. Nella CPU è integrata la memoria cache che è mirata a rendere immediatamente disponibili le prossime istruzioni da eseguire, è una memoria velocissima, più memoria cache c'è in un PC, più alte sono le sue prestazioni. Questo tipo di memoria non è espandibile.

La CPU è sostanzialmente costituita dalle seguenti componenti:

- La **ALU** (unità logico aritmetica) che esegue le operazioni aritmetiche, le operazioni logiche.
- La **CU** (control unit) che sovrintende al coordinamento delle varie componenti hardware del microprocessore durante l'esecuzione dei processi.
- I **registri**, sui quali la CPU memorizza in modo temporaneo i dati da elaborare.

**Le memorie**, sono anch'esse di varie tipologie e svolgono diverse funzioni:

**RAM** (Random Access Memory): La memoria RAM è una memoria volatile, cioè perde il suo contenuto nel momento in cui non viene più alimentata dalla tensione di lavoro. Tutto il lavoro che viene introdotto nell'elaboratore, va a depositarsi sulla memoria ram fino a quando non si effettua il salvataggio sull'hard disk o su un'altra memoria permanente. Se dovesse spegnersi il computer senza avere salvato il lavoro, i dati sarebbero irrimediabilmente persi. Insieme alla CPU è uno dei dispositivi che contribuisce ad aumentare le prestazioni del computer, la ram è in genere espandibile entro dimensioni dipendenti dall'architettura della stessa macchina.



Hardware Software,  
CPU, Alu e Cu,  
spiegati in modo  
semplice da due  
ingegneri di Amazon e  
MS XBOX



**ROM** (Read Only Memory): la memoria ROM è una memoria di sola lettura nella quale il costruttore del PC memorizza un programma di inizializzazione e controllo, il **BIOS**, che all'accensione va in esecuzione nella cosiddetta fase di bootstrap, durante la quale vengono effettuati una serie di test e riconosciute le componenti installate come la memoria ram, le unità di memoria di massa come i dischi rigidi, le periferiche come la tastiera. Se in questa fase si verifica qualche anomalia, questa viene segnalata con una serie di beep che codificano il tipo di errore riscontrato, altrimenti si procede con il caricamento del sistema operativo.

La ROM non è una memoria volatile, allo spegnimento del PC i dati memorizzati sulla ROM non vengono persi poiché sono registrati su questa memoria in modo permanente. Il costruttore può anche rilasciare versioni aggiornate e migliorate del bios che possono essere aggiornate con una particolare e delicata procedura nella versione delle ROM riscrivibili, le **EEPROM** acronimo di *Electrically Erasable*, ROM.



**L'HARD DISK** o disco rigido è un supporto magnetico dove vengono installati i programmi e dove possono essere conservati e salvati i nostri lavori. Può essere costituito da un insieme di dischi, gira a una velocità che varia secondo la capacità di memorizzazione. Sul disco si legge e si scrive mediante una testina di lettura-scrittura. Esistono in commercio dischi esterni che si collegano al computer mediante porta USB. L'attuale tecnologia inizia a essere sostituita dalla cosiddetta memoria a stato solido (memoria flash) SSD.



**I DISCHI OTTICI** I dischi ottici sono supporti dove tramite un raggio laser vengono scritte o lette informazioni. La prima tecnologia prevedeva solo dischi di sola lettura, l'utente può solo leggere i dati, ma non scriverci sopra.

Poi sono arrivati i dischi sui quali è possibile scrivere una volta sola e infine i dischi sui quali si può scrivere più volte (Re-Writable). Su alcuni DVD la registrazione può avvenire in entrambe le facciate. Vengono utilizzati soprattutto per archiviare video e prodotti multimediali.



**LA PEN DRIVE, MEMORIA FLASH O CHIAVETTA USB** è un tipo di memoria di massa, molto utilizzata, portatile, viene collegata al computer tramite la porta USB. I dati vengono memorizzati nella cosiddetta memoria flash. L'elaboratore identifica la presenza di una chiavetta con la dicitura *disco rimovibile*. Questo tipo di periferica appartiene alla categoria plug&play, che significa collega e usa, cioè non richiede in genere installazione e può essere collegata senza dover spegnere il computer.



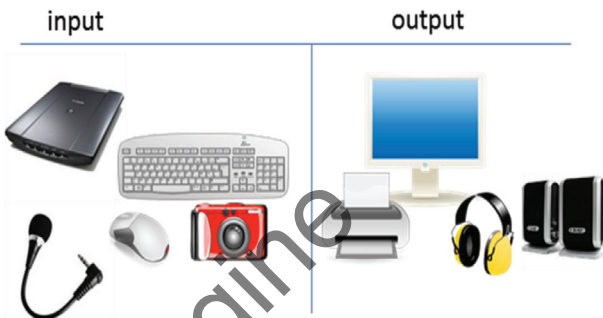
**LE SCHEDE SD** Le memorie SD (Secure Digital) permettono di conservare informazioni in formato digitale. Oggi sia i PC, che i laptop oltre ai telefonini, alle fotocamere e videocamere, hanno appositi drive, lettori che ci consentono l'utilizzo di queste schede in lettura e scrittura, A seconda delle dimensioni abbiamo tre formati di schede: la SD (32x24 mm), la mini SD (21,5x20 mm) e la micro SD (11x15 mm).



I dispositivi interni al Case ricevono sequenze d'istruzioni dall'esterno attraverso unità periferiche di input e comunicano con l'esterno attraverso unità periferiche di output.

Tra le periferiche d'inserimento dei dati (**input**), troviamo la tastiera, il mouse, il touchpad, il touchscreen, lo scanner, il joystick, il microfono, la webcam.

Tra le periferiche di uscita dei dati (**output**), troviamo il monitor, la stampante, le cuffie, plotter, sintetizzatori vocali, ecc.






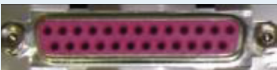


Le componenti del PC



Tali periferiche possono essere collegate al computer tramite **Le porte input output**.



Esaminiamone alcune:

	<p>✓ La porta USB permette di connettere diversi tipi di dispositivi tra cui pen drive e dischi esterni ed anche modem, per la connessione a internet, garantendo velocità di trasferimento dati adeguate alle necessità.</p>
	<p>✓ La porta firewire è stata utilizzata per trasferire flussi multimediali, in genere da videocamere analogiche, perché più veloci degli standard delle porte usb di quel tempo.</p>
	<p>✓ Porta seriale, in cui i dati vengono inviati un bit alla volta. È sempre più rara trovarla nei nuovi pc in quanto quasi tutte le periferiche sono prodotte con connessione USB.</p>
	<p>✓ La porta parallela si usa in genere per collegarci la stampante – I connettori della porta parallela hanno 25/36 pin. Anche in questo caso le periferiche che utilizzavano questo standard stanno migrando verso la tecnologia USB</p>
	<p>✓ Porta PS/2 serve per collegare il mouse e la tastiera.</p>
	<p>✓ Porta porta ethernet (LAN) che consente di collegare il computer ad una rete locale o ad un router, attraverso un cavo particolare è possibile collegare direttamente due computer</p>



Oggi la tendenza della produzione si sta orientando verso periferiche sempre più comode ed efficienti. Poter essere svincolati da un collegamento fisico, cioè da un cavo, è una grande comodità ed è per questo che si utilizzano sempre più spesso tastiere e mouse senza fili, come anche stampanti e casse acustiche. Il canale non è più un cavo fisico ma l'etere, la trasmissione si definisce semplicemente WIRELESS (senza cavo), il segnale viene codificato e trasmesso mediante onde radio o infrarossi.

Guarda il video con una sintesi dei contenuti di questo paragrafo



## Virus e Antivirus

Un virus è un programma che può infettare file e si può riprodurre e diffondere da un sistema informatico ad un altro provocando danni più o meno gravi, che vanno dal semplice rallentamento di funzionamento fino alla cancellazione di file e al danneggiamento dell'hard disk, altri tipi di danni anche più gravi sono i furti di dati, codici di accesso e password che possono consentire addirittura transazioni finanziarie illecite a insaputa dell'utente.

Vari sono i metodi di diffusione del virus, in passato la contaminazione poteva avvenire solo caricando sul proprio computer da dischetto qualche file infetto, oggi ciò succede anche più frequentemente per il massiccio utilizzo delle pen-drive e delle reti.

I virus si diffondono anche attraverso la posta elettronica, in genere con gli allegati. Quando un utente apre l'allegato di una mail ricevuta, se questa contiene in virus, infetta il pc. Gli allegati a volte hanno frasi invitanti che possono essere di vario genere e possono arrivare anche da mittenti a noi conosciuti, ma a loro insaputa, l'apertura dell'allegato corrisponde in genere all'esecuzione del file infetto e quindi alla contaminazione.

Anche tramite il download di software da siti internet non sicuri, si può essere contaminati da virus. Per prevenire l'infezione da virus, si raccomanda di **installare un software antivirus che deve essere aggiornato con frequenza**.

Per prevenire perdite di dati, a causa di virus o per malfunzionamento del PC, è consigliabile **periodicamente** creare una copia di **backup** dei dati, che deve essere fatta su supporti rimovibili come ad esempio un hard disk esterno, un cd o DVD, una chiavetta USB e tenuta in un posto sicuro.

## Salute, sicurezza e ambiente

Tutti coloro che lavorano con un PC devono sapere che una postura scorretta, associata alla mancanza di pause nel lavoro, può condurre a disturbi dell'apparato muscoloscheletrico.

La postazione di lavoro deve rispondere a determinate caratteristiche, come definito nell'allegato VII del Decreto legislativo 19 settembre 1994 - n. **626**, al fine di non influire negativamente sulla salute del lavoratore.

**L'Ergonomia** è la disciplina scientifica che si occupa dei problemi relativi al lavoro umano integrando le ricerche e le soluzioni offerte da varie altre discipline come la medicina generale e la medicina del lavoro, la fisiologia, la psicologia, la sociologia e la fisica, al fine di realizzare un adattamento ottimale del sistema uomo-macchina-ambiente di lavoro alle capacità e ai limiti psico-fisiologici dell'uomo.

Le soluzioni ai problemi di ergonomia sono diverse per ogni singolo componente la postazione di lavoro, per esempio per la **sedia**, la migliore scelta, dal punto di vista ergonomico, è una seduta regolabile per adattare al meglio la postura, tale da poter stare seduti in posizione eretta o leggermente inclinata all'indietro, inoltre è necessario che la parte inferiore della schiena sia completamente a contatto dello schienale che deve sostenerla. Le spalle devono essere diritte, le braccia distese, vicino al corpo, con i gomiti ad angolo retto. Gli avambracci e le mani devono essere paralleli al piano di lavoro.

**Il monitor**, posizionato in modo scorretto può causare dolori al collo, alle spalle e affaticamento della vista a causa dei riflessi. Esso deve essere leggermente inclinato verso l'alto, la parte superiore dello schermo deve trovarsi allo stesso livello degli occhi, finestre e fonti luminose dovrebbero essere poste lateralmente al video. La distanza degli occhi varia in base alle dimensioni schermo e può essere calcolata moltiplicando la dimensione in pollici per quattro, il risultato è la distanza in centimetri, uno schermo 19 pollici richiede una distanza ottimale di circa 76 centimetri (19 x 4). La visione continuata del monitor è da evitare, circa ogni venti minuti è opportuno mettendo a fuoco un punto lontano per alcuni secondi.

Una buona norma per la tutela della salute di chi utilizza il personal computer giornalmente e per varie ore settimanali, è fare una pausa di almeno 15 minuti ogni due ore, facendo anche qualche esercizio fisico di stretching, è anche consigliabile di aerare regolarmente i locali specie nel caso in cui siano installati più apparecchi.

Una buona norma di sicurezza è di fare attenzione alle prese di corrente e ai fili elettrici, non inserire nella stessa presa vari cavi di alimentazione poiché potrebbe causare un sovraccarico di tensione.

Per salvaguardare l'ambiente, si consiglia di stampare solo se necessario, le nuove tecnologie permettono, infatti, di trasferire e far circolare facilmente le informazioni in formato elettronico, inoltre si raccomanda di riciclare e smaltire correttamente le apparecchiature e i materiali utilizzati.



## 2. Codifica dell'informazione



Autori: **Alessandra D'Orazio, Giorgia Martina, Paolo Lillo**  
 Rielaborazione e riduzione a cura di: **Angelo Oliva, Salvatore Madaro**

Competenze	Abilità	Conoscenze
Sapere codificare un'informazione numerica	Riconoscere i diversi tipi di informazioni	Dato e informazione
Saper codificare un'informazione alfanumerica		Informazione Analogica e digitale
Saper individuare le tecniche di codifica conformi ai dati coinvolti		Bib e byte
		Codifica dei dati

### La codifica dell'informazione, il bit

Diamo una definizione di "INFORMATICA":

**"L'informatica è la disciplina che si occupa della *codifica*, della memorizzazione, della trasmissione e dell'elaborazione dell'INFORMAZIONE"**



Intendiamo per **INFORMAZIONE**

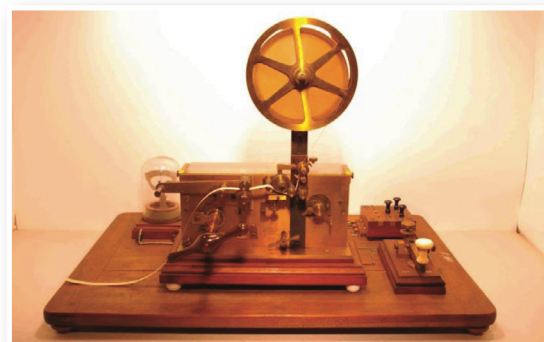
"Ogni messaggio o elemento conoscitivo che è possibile comunicare o acquisire".

La parola e le espressioni del corpo, i segni (dai graffiti preistorici alla scrittura) e in generale tutto ciò che può essere percepito dai sensi, hanno permesso all'uomo nel corso della storia di **comunicare**, cioè di trasferire l'informazione non solo da uomo a uomo ma anche da generazione a generazione. I modelli culturali di oggi sono il risultato della lunga evoluzione del processo.

Se la comunicazione è alla base del trasferimento delle informazioni, il ragionamento e quindi l'uso della capacità logica-deduttiva hanno da sempre avuto il compito di elaborare l'informazione attraverso l'uso di modelli interpretativi che con lo scorrere del tempo sono divenuti sempre più articolati e raffinati.

L'uomo, del resto, si è sempre ingegnato per superare i suoi limiti utilizzando tecniche e strumenti per rendere sempre più efficace la sua capacità di comunicare: dall'uso dei tam-tam e dei segnali di fumo alle moderne tecnologie.

Il lettore avrà certamente sentito parlare del "codice Morse": quando nel 1838 si realizzarono i primi telegrafi via cavo (e successivamente via radio) gli unici "segnali" che era possibile produrre e distinguere in modo chiaro e immediatamente comprensibile erano due segnali acustici di lunghezza lunga e breve, rispettivamente linea e punto; la produzione dei segnali avveniva da un lato della linea telegrafica (trasmettitore), agendo su un interruttore a molla: **ad una pressione breve corrispondeva un segnale del tipo punto mentre ad una pressione lunga faceva seguito un segnale del tipo linea**; dall'altro lato della linea telegrafica il segnale elettrico ricevuto attivava un elettromagnete, il cui movimento portava in contatto una penna su una striscia di carta trascinata da un meccanismo a orologeria.



Il messaggio più famoso del codice Morse è certamente SOS, il segnale di richiesta di soccorso ("Save Our Souls" cioè "Salvate le nostre anime") corrispondente alla sequenza:



dove i primi tre punti stanno per "S", le tre linee stanno per "O" e gli ultimi tre punti stanno nuovamente per "S". Punto e linea rappresentano quindi i 2 simboli grazie ai quali è possibile rappresentare i messaggi dell'alfabeto Morse.

Quindi, una volta stabilita la corrispondenza tra l'insieme di punti e linee e le lettere dell'alfabeto, le cifre, i caratteri di interpunzione e alcuni segni convenzionali (ovvero una volta stabilita la **codifica** nell'alfabeto Morse di ogni singolo carattere che compone un testo scritto), un qualsiasi brano può essere codificato utilizzando punti e linee; d'altro canto, chiunque abbia a disposizione una tabella di conversione

punti-linee → carattere

sarà in grado di leggere il testo originale.

A	· - -	J	· - - - -	S	· · ·
B	- · · ·	K	- · - -	T	-
C	- - - ·	L	· - · ·	U	· · - -
D	- · · ·	M	- - -	V	· · · -
E	·	N	- · ·	W	· - - -
F	· · · -	O	- - - -	X	- · · ·
G	- - · ·	P	· - · ·	Y	- · - -
H	· · · ·	Q	- - - ·	Z	- · - ·
I	· ·	R	· - · ·		

L'alfabeto Morse è un esempio di **CODIFICA**; è chiaro che per la rappresentazione codificata dell'informazione è necessaria una **CONVENZIONE** (detta anche **PROTOCOLLO**), cioè un accordo tra le parti che devono comunicare sul **SIGNIFICATO** da dare alle sequenze di simboli: tanto più diffusa è tale convenzione tanto più estesa sarà la platea dei possibili fruitori.

L'introduzione dell'informatica ha avviato un'epoca in cui l'informazione e il trattamento dell'informazione, grazie all'evoluzione della tecnologia elettronica, hanno assunto forme sempre più automatiche. Al fine di potere essere "gestita" da

processi automatici è stato indispensabile rendere l'informazione sempre più autonoma cioè indipendente dalla capacità interpretativa dell'uomo.

Prima di entrare nel vivo della trattazione sulla codifica dei più comuni tipi di informazione (spiegheremo tra un po' cosa intendiamo per "tipi"), soffermiamo la nostra attenzione sulla definizione e sulla codifica di un'informazione in ambiente più specificatamente informatico ponendoci due domande.

**Prima domanda: se l'informazione è l'oggetto intorno a cui ruota la disciplina Informatica, che cosa è un dato e che relazione intercorre tra un dato e un'informazione?**

I termini **dato** e **informazione** sono a volte utilizzati come sinonimi, ma i due concetti non coincidono. Facciamo alcuni esempi:

- il numero intero 16: è un dato o un'informazione? È un dato, diventa un'informazione se aggiungo che 16 è l'età di una ragazza oppure che 16 sono gli amici che ho invitato alla mia festa.
- Maria ha i capelli rossi: è un dato o un'informazione? È un'informazione: il dato, colore rosso, è "diventato" un'informazione in quanto associato ai capelli di Maria.

Quindi: un dato è una parte dell'informazione, la quale a sua volta può essere considerata come un dato a cui è associato il significato che ha nel contesto di studio. Più precisamente, possiamo considerare l'informazione costituita da tre elementi:

1. il valore (il dato)
2. il tipo (carattere, numero intero, stringa,..) utilizzato per esprimere il valore
3. il significato (la semantica) da associare al valore

Se l'informatica si occupa, tra le varie cose, così come detto nella definizione iniziale, di memorizzare ed elaborare le informazioni, che cosa verrà trascritto ovvero memorizzato e che cosa verrà elaborato? La risposta è: **i dati!**

**Seconda domanda: nel campo dell'informatica, come si può memorizzare un'informazione e in quanti modi diversi è possibile rappresentare la stessa informazione?** Partiamo dal seguente ragionamento: i dati sono memorizzati nelle memorie (di massa, centrale, RAM, ROM, ..) ed elaborati dai computer:

memorie e computer sono supporti fisici costituiti da dispositivi in grado di distinguere tra due (e solo due) diversi valori di riferimento di una grandezza fisica (assenza/presenza di tensione, luminoso/non luminoso). Tali dispositivi sono detti elettricamente bistabili.

Se associamo a ciascun valore un simbolo, possiamo rappresentare i due diversi stati del dispositivo in esame con due simboli distinti; i due simboli utilizzati sono {0, 1}, che costituiscono l'**alfabeto binario**. Attenzione: 0 e 1 non sono numeri ma simboli, da non confondere quindi con i primi due numeri naturali, vedremo a breve come con questi due valori è possibile codificare informazione di vario tipo.

Possiamo ora definire il concetto informatico di **bit** (binary Digit = cifra binaria)

**“Il bit è l’unità elementare di informazione cioè la minima quantità di informazione codificabile”**



Essendo indispensabili, come appena visto, almeno due simboli per rappresentare un’informazione, potremmo ridefinire il bit come **uno tra i DUE simboli** possibili adottati nello specifico contesto binario (‘punto’ e ‘linea’ nell’alfabeto Morse oppure ‘0’ e ‘1’ nel sistema numerico binario). L’uso di un alfabeto binario (cioè basato sull’adozione di un alfabeto di due simboli) è largamente utilizzato nella tecnologia in virtù della facile e quindi economica riproducibilità nei vari contesti della fisica (elettricità, magnetismo, ottica) di **uno tra due stati fisici**: tensione/nontensione (elettricità), magnetizzato/smagnetizzato (magnetismo), luminoso/non luminoso (ottica).

**Il bit vale 0 se è NO, FALSO, SPENTO, NON PASSA CORRENTE.**

**Il bit vale 1 se è SI’, VERO, ACCESO, PASSA CORRENTE.**

Il bit è quindi il più piccolo dato memorizzabile in un elaboratore e può rappresentare una sola informazione binaria. Qualsiasi altro tipo di informazione è rappresentato come sequenze finite di 0 e 1; come questi 2 simboli sono combinati tra loro dipende dal codice usato, ovvero dalla relazione che trasforma ogni dato in una stringa di bit.

Ma una sequenza di bit cosa rappresenta? A quale possibile informazione è associata?

Nei paragrafi successivi daremo una risposta a questi quesiti; più precisamente affronteremo il problema di come è possibile codificare.

- numeri
- caratteri
- immagini
- suoni
- video

*Suggerimento: guarda il video collegato al QR CODE man mano che affronti le varie codifiche di dati: numeri, video immagini e suoni, guardalo anche al termine del capitolo, probabilmente tutto risulterà più chiaro.*



## Codifica dei numeri

Iniziamo con il ricordare alcune definizioni:

- **Numero**: oggetto o ente astratto
- **Numerale**: insieme di simboli che rappresenta un numero in un dato sistema di numerazione
- **Sistema di numerazione**: un insieme di simboli e un insieme di regole; i simboli servono per rappresentare un numero, le regole stabiliscono come scrivere e come operare con i numerali.

Di conseguenza possiamo rappresentare un numero e scrivere il suo numerale solo dopo aver stabilito il sistema di numerazione che intendiamo utilizzare: se cambiamo sistema di numerazione, cambia la rappresentazione del numero.

Esempio: il numerale 15 nel sistema decimale diventa XV nel sistema di numerazione romano.

Esaminiamo allora due caratteristiche del sistema di numerazione che usiamo correntemente, il sistema decimale:

- è in base 10: ogni numero è rappresentato da una combinazione di 10 simboli diversi {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- è posizionale: esiste una regola che stabilisce un legame tra il simbolo e la sua posizione nel numerale che rappresenta il numero, cioè ogni cifra assume un valore diverso a seconda della posizione occupata nella rappresentazione del numero.

Esempio: numerale = 184  $\rightarrow 1 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0 = 100 + 80 + 4$

posizione  $\rightarrow$  210

(N.B. Il simbolo di moltiplicazione è indicato con “\*” e non con “x”, in conformità con la simbologia adottata in ambiente informatico)

Come si può osservare, la posizione delle cifre in un numerale è tanto più importante, perché è quella che più contribuisce al valore finale (in gergo “più significativa”), quanto più a sinistra si trova nella sequenza numerica: la cifra MENO SIGNIFICATIVA è quella più a destra, che nel sistema decimale rappresenta le “unità”.

Il sistema decimale probabilmente deve ad una proprietà anatomica (le 10 dita delle mani) la sua diffusione, ma non risulta altrettanto vantaggioso quando l'elaborazione dei numeri si sposta dall'uomo al computer.

Come già accennato nel paragrafo precedente, un computer in quanto macchina "sente" 2 diversi stati (acceso/spento, presenza/assenza di tensione); conseguenza quasi ovvia: per codificare i numeri che devono essere elaborati da un computer si utilizza il sistema di numerazione in base 2, ovvero il sistema di numerazione posizionale che ha il bit come elemento di base.

Prenderemo in esame altri 2 sistemi di numerazione posizionali, il sistema di numerazione in base 8 (ottale) e il sistema in base 16 (esadecimale), entrambi caratterizzati dal fatto che le basi sono potenze della base 2 (spiegheremo in seguito cosa comporta questa considerazione).

Due annotazioni:

- useremo il termine numero anche quando sarebbe più corretto utilizzare numerale
- ogni sequenza di simboli che scriveremo avrà come pedice la base in cui è espresso il numero.

Esempio:  $215_{10}$  rappresenta un numero espresso in base 10

$1011_2$  rappresenta un numero espresso in base 2 (si legge **uno zero uno uno**)

### Sistema di numerazione in base 2

Il sistema binario utilizza due simboli, 0 e 1, grazie ai quali possiamo rappresentare qualsiasi numero, purché si abbia a disposizione un numero di bit sufficienti. Osserviamo la seguente una tabella in cui riportiamo tutti i valori rappresentabili con un determinato numero di bit

In generale con n bit si possono rappresentare interi positivi compresi nell'intervallo  $[0, 2^n - 1]$ .

Attenzione: una volta che è stato fissato n, occorre rappresentare anche gli 0 non significativi.

### Sistema di numerazione in base 8

- è un sistema di numerazione posizionale
- ogni numero è espresso come combinazione degli 8 simboli: 0, 1, 2, 3, 4, 5, 6, 7

### Sistema di numerazione in base 16

- è un sistema di numerazione posizionale
- ogni numero è espresso come combinazione dei 16 simboli:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Scriviamo i primi 21 numeri naturali in base 10, in base 2, in base 8, in base 16 (per semplicità, non ci preoccupiamo della lunghezza delle rappresentazioni nelle varie basi).

	base 10	base 2	base 8	base 16
	0	0	0	0
	1	1	1	1
	2	10	2	2
	3	11	3	3
	4	100	4	4
	5	101	5	5
	6	110	6	6
	7	111	7	7
	8	1000	10	8
	9	1001	11	9
	10	1010	12	A
	11	1011	13	B
	12	1100	14	C
	13	1101	15	D
	14	1110	16	E
	15	1111	17	F
	16	10000	20	10
	17	10001	21	11
	18	10010	22	12
	19	10011	23	13
	20	10100	24	14

Ma allora: se non abbiamo a disposizione la tabella precedente, come possiamo affermare che la sequenza 1101 in base 2 equivale a 13 in base 10 ( $1101_2=13_{10}$ )?

E soprattutto: come procediamo per valori maggiori di 20?

**Conversione da base 10 a base b: metodo delle divisioni successive**

(Per numeri interi senza segno)

Esiste una tecnica semplice e intuitiva che permette di convertire un numero espresso in base 10 nell'equivalente espresso in un'altra b qualsiasi.

Questo metodo consiste nell'applicare una sequenza istruzioni al numero che si vuole convertire in una determinata base, partiamo con un esempio, Supponiamo di voler convertire il numero  $N=106_{10}$  nell'equivalente numero in base  $b=2$ ,

1. eseguire la divisione intera **N:2** (nel nostro esempio  $106:2$ )
2. **r**, il resto ottenuto (nel nostro esempio  $r=0$ ) è la prima cifra (quella più a destra) di quello che sarà il nostro numero binario,
3. **Fino a quando** il quoziente ottenuto è maggiore di zero  $q>0$  (nel nostro esempio  $53>0$ ) **si ripete** la divisione **q:2** ( $53:2$ ) ottenendo un **nuovo resto r** (nel nostro esempio 1) da scrivere a sinistra del resto precedente e un **nuovo quoziente q** (nel nostro esempio 26)

$$\begin{array}{r}
 106:2 = 53 \quad \text{resto } 0 \\
 53:2 = 26 \quad \text{resto } 1 \\
 26:2 = 13 \quad \text{resto } 0 \\
 13:2 = 6 \quad \text{resto } 1 \\
 6:2 = 3 \quad \text{resto } 0 \\
 3:2 = 1 \quad \text{resto } 1 \\
 1:2 = 0 \quad \text{resto } 1
 \end{array}$$

al termine avremo scritto una successione di resti da destra verso sinistra che corrispondono al risultato  $1101010_2$

1101010

Proviamo ad applicare il metodo per convertire lo stesso numero  $N= 106_{10}$  nella base  $b=8$

$106 : 8 = 13 \quad \text{resto } 2$
$13 : 8 = 1 \quad \text{resto } 5$
$1 : 8 = 0 \quad \text{resto } 1$
<b>risultato <math>152_8</math></b>

è immediato notare come la stessa sequenza di istruzioni funziona anche per la base 8 e per sistemi di numerazione in altre basi, bisogna fare attenzione però per i sistemi di numerazione con base maggiore di 9, quello di nostro interesse è il sistema in base 16, come potrei distinguere nella sequenza 1210 se ho avuto due resti cioè prima 12 e poi 10, o tre resti 1, 2, 10 oppure 12, 1, 0 oppure quattro resti 1,2,1,0 ? la soluzione è quella di usare altri simboli per i resti maggiori di nove e cioè:

A	B	C	D	E	F
10	11	12	13	14	15

Trasformiamo ad esempio il numero  $N= 463_{10}$  nella base  $b=16$ :

Possiamo quindi affermare che:

$463 : 16 = 28 \quad \text{resto } 15 \rightarrow \mathbf{F}$
$28 : 16 = 1 \quad \text{resto } 12 \rightarrow \mathbf{C}$
$1 : 16 = 0 \quad \text{resto } 1$
<b>risultato <math>1CF_{16}</math></b>

La **conversione** di un numero da base 10 a base b, che sia 2, 8, 16 o altro si effettua col **Metodo delle divisioni successive**

Clicca o inquadra il QR CODE per un video esempio di conversione da base 10 a base b col metodo delle divisioni successive







Lo sapevi? Puoi controllare i risultati delle conversioni anche con la calcolatrice di windows, basta scegliere dal menù la modalità programmatore, scegliere la base tra HEX esadecimale, DEC decimale, OCT ottale, BIN binario, digitare il numero e vedrai immediatamente il risultato nelle altre tre basi, noterai che scegli binario che tutti i tasti delle altre cifre e delle lettere risulteranno disabilitati.

Se invece abbiamo un numero in base 2, 8 o 16 e vogliamo conoscere l'equivalente decimale?

### **Conversione da base b a base 10: metodo della forma polinomiale**

I sistemi di numerazione posizionale hanno in comune la possibilità di essere rappresentati in forma polinomiale, ogni cifra assume un valore diverso a seconda della posizione occupata nella rappresentazione del numero. Prendiamo il numero  $284_{10}$  osserviamo la seguente tabella in cui abbiamo inserito ogni cifra partendo da destra, che corrisponde alla **posizione 0**.

cifra	2	8	4
posizione	2	1	0
peso	$10^2$	$10^1$	$10^0$

Se eleviamo la base 10 al valore della posizione otteniamo il peso:

- $10^0 = 1$  cioè unità e 4 sono le unità del numero  $284_{10}$ , cioè  $4 \times 1 = 4$
- $10^1 = 10$  cioè decine e 8 sono le decine del numero  $284_{10}$ , cioè  $8 \times 10 = 80$
- $10^2 = 100$  cioè centinaia e 2 sono le centinaia del numero  $284_{10}$ , cioè  $2 \times 100 = 200$

quindi sommando i valori ciascuna cifra rappresenta abbiamo:  $200 + 80 + 4 = 284$

riassumendo  $284_{10} \rightarrow 2 \times 10^2 + 8 \times 10^1 + 4 \times 10^0 = 200 + 80 + 4 = 284$

l'operazione sembra avere poco senso perché siamo partiti da un numero in base 10, proviamo a usare la stessa forma per il numero  $(11011)_2$ :

cifra	1	1	0	1	1
posizione	4	3	2	1	0
peso	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

Se eleviamo la base 2 al valore della posizione otteniamo il peso:

- $2^0= 1$       cioè la prima cifra per il suo peso vale       $1 \times 1 = 1$
- $2^1= 2$       cioè la seconda cifra per il suo peso vale       $1 \times 2 = 2$
- $2^2= 4$       cioè la terza cifra per il suo peso vale       $0 \times 4 = 0$
- $2^3= 8$       cioè la quarta cifra per il suo peso vale       $1 \times 8 = 8$
- $2^4= 16$      cioè la quinta cifra per il suo peso vale       $1 \times 16 = 16$

quindi sommando i valori ciascuna cifra rappresenta abbiamo:

$$16 + 8 + 0 + 1 + 1 = 27$$

il numero 27 è espresso in base 10 quindi riassumendo

$$(11011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 8 + 0 + 1 + 1 = 27 \text{ (in base 10)}$$

Possiamo quindi affermare che:

La **conversione** di un numero **N** da **base b** che sia **2, 8, 16** o altro a **base 10**, si effettua con la **forma polinomiale**:

1. si moltiplica ogni cifra del numero **N** per il valore della base elevato ad esponente pari alla posizione che ha la cifra nella sequenza di simboli
2. si sommano i valori così ottenuti

Clicca o inquadra il QR CODE per un video esempio di conversione da base b a base 10 col metodo della forma polinomiale ed una tecnica abbreviata solo per conversione 2 - 10.



**Caso particolare: conversioni dirette per basi (2, 8, 16) uso delle tabelle di conversione**

Supponiamo di dover convertire il numero  $1110101000100_2$  nella base otto, con i metodi che abbiamo visto potremmo farlo **eseguendo due conversioni**, dovremmo prima trasformare il numero **dalla base 2 alla base 10 (forma polinomiale) e poi** far seguire la conversione **da base 10 a base 8 (divisioni successive)**.

Esiste però un metodo molto più semplice, che prevede un unico procedimento, possibile grazie ad una particolare relazione tra le basi di partenza e di arrivo, cioè che l'una è potenza dell'altra o viceversa, cioè in questo caso l'identità  $2^3=8$  che permette di convertire ogni gruppo di 3 bit in un simbolo in base 8 e, viceversa, ogni simbolo in base 8 nella sequenza di 3 bit in base 2.

Per velocizzare tali operazioni utilizziamo una tabella nella quale ad ogni cifra del sistema in base 8 associamo l'equivalente rappresentazione binaria usando tre cifre binarie (bit).

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

A questo punto dal numero  $1110101000100_2$  isoliamo gruppi di tre cifre partendo da destra e scriviamo la cifra ottale corrispondente

$$\begin{array}{cccccc}
 001 & 110 & 101 & 000 & 100 & \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
 1 & 6 & 5 & 0 & 4 & 
 \end{array}$$

All'ultimo gruppo a sinistra si aggiungono eventualmente gli zeri necessari per arrivare a tre cifre quindi otteniamo che:  $1110101000100_2 = 16504_8$ .

La conversione del numero  $2360_8$  utilizzando la tabella ci darebbe  $2=010$   $3=011$   $6=110$   $0=000$  quindi  $010011110000_2$  dal cui risultato va cancellato lo zero più a sinistra essendo cifra non significativa.



Analogamente si procede per trasformare il numero **dalla base 2 alla base 16 e viceversa**.

La relazione tra le basi di partenza e di arrivo, in questo caso è l'identità  $2^4=16$  che permette di convertire ogni gruppo di 4 bit in un simbolo in base 16 e, viceversa, ogni simbolo in base 16 nella sequenza di 4 bit in base 2.

A questo punto dal numero  $1110101010100_2$  isoliamo gruppi di tre cifre partendo da destra e scriviamo la cifra ottale corrispondente

0001	1101	0101	0100
↓	↓	↓	↓
1	D	5	4

All'ultimo gruppo a sinistra si aggiungono eventualmente gli zeri necessari per arrivare a quattro cifre quindi otteniamo che

$$1110101010100 = 1D54_{16}$$

La conversione del numero  $3B7A_8$

utilizzando la tabella ci darebbe  $3 = 0011$   $B = 1011$   $7 = 0111$   $A = 1010$  quindi

$0011101110111010_2$  dal cui risultato vanno cancellati gli zeri più a sinistra essendo cifre non significative.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

**ATTENZIONE:** al termine di questo capitolo troverai un'attività di laboratorio realizzata con i fogli elettronici, in questo caso con Openoffice Calc, in cui viene illustrato come realizzare conversioni da base b a base 10 e conversioni da base 10 a base b.

*Clicca o inquadra il QR CODE per un video esempio di conversione diretta con l'uso delle tabelle e tecnica abbreviata per ricordare come costruire le stesse tabelle.*



### Esercizi: conversioni di base

a) da base 10 a base 2

1) 23	[10111]
2) 38	[100110]
3) 100	[1100100]
4) 32	[100000]
5) 70	[1000110]
6) 65	[1000001]
7) 98	[1100010]
8) 255	[11111111]
9) 318	[100111110]
10) 512	[1000000000]

b) da base 2 a base 10

11) 1011	[11]
12) 10110	[22]
13) 110011	[51]
14) 111111	[63]
15) 1011100	[92]
16) 11000000	[192]
17) 111010101	[469]
18) 100000000	[256]
19) 10000011	[131]
20) 110011001	[409]

c) da base 10 alla base indicata

21) 100 in base 8	[144 <sub>8</sub> ]
22) 1024 in base 16	[400 <sub>16</sub> ]
23) 38 in base 8	[46 <sub>8</sub> ]
24) 153 in base 16	[99 <sub>16</sub> ]

d) da base N a base 10

25) 152 <sub>8</sub>	[106]
26) A1F <sub>16</sub>	[2591]
27) 100 <sub>16</sub>	[256]
28) 181 <sub>8</sub>	[impossibile]

e) da base N a base M

29) 615 <sub>8</sub> in base 2	[110001101 <sub>2</sub> ]
30) 326 <sub>8</sub> in base 2	[11010110 <sub>2</sub> ]
31) 1110110 <sub>2</sub> in base 16	[76 <sub>16</sub> ]
32) 11110110 <sub>2</sub> in base 8	[366 <sub>8</sub> ]
33) 1AB6 <sub>16</sub> in base 2	[1101010110110 <sub>2</sub> ]
34) ABC <sub>16</sub> in base 2	[101010111100 <sub>2</sub> ]



**Approfondimento: Operazioni con i numeri binari**

Il sistema di numerazione binario è, come sottolineato più volte, un sistema di numerazione posizionale: di conseguenza in esso le quattro operazioni fondamentali seguono le stesse regole applicate nel sistema decimale.

(Per semplicità, supponiamo che non ci siano limiti circa il numero di bit che occorrono per rappresentare il risultato).

**Addizione:** si sommano le cifre di uguale peso, ricordando che si può avere un riporto (*carry*) sul bit di peso immediatamente superiore.

+	0	1
0	0	1
1	1	0

← con riporto di 1

**Sottrazione:** si sottraggono le cifre di uguale peso, ricordando che si può avere un prestito (*borrow*) sul bit di peso immediatamente superiore

(\*) sulla prima colonna a sinistra le cifre del minuendo, sulla prima riga le cifre del sottraendo

-	0	1
(*)		
0	0	1
1	1	0

← con prestito di 1

**Moltiplicazione:** le cifre costituenti il primo numero (moltiplicando) vengono moltiplicate per ciascuna cifra del secondo numero (moltiplicatore): se la cifra del moltiplicatore

*	0	1
0	0	0
1	0	1

è pari a 1 si riscrive il moltiplicando, in caso contrario (cifra=0) il risultato è pari a 0. Si sommano tutti i numeri ottenuti (attenzione ai riporti!)

**Divisione:** anche in questo caso il procedimento coincide con quello applicato tra numeri in base 10, con la semplificazione che il gruppo di cifre del dividendo esaminato può contenere il divisore 1 o 0 volte.

**Addizione:**  $110100_2 + 111000_2$

**Sottrazione:**  $10011_2 - 1111_2$

Esempi:

$$\begin{array}{r}
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \hline
 1 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1}
 \end{array}$$

$$\begin{array}{r}
 \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \hline
 0 \phantom{0} \phantom{0} \phantom{1} \phantom{0} \phantom{0}
 \end{array}$$

$110100_2 + 111000_2 = 1100100_2$

$10011_2 - 1111_2 = 100_2$

**Moltiplicazione:**  $11101_2 * 1110_2$

**Divisione:**  $110101_2 : 11_2$

$$\begin{array}{r}
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \hline
 1 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1}
 \end{array}$$

$$\begin{array}{r}
 1 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 1 \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \phantom{1} \\
 \hline
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 \hline
 0 \phantom{0} \phantom{0} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 \hline
 0 \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 0 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 \hline
 1 \phantom{0} \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 1 \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \\
 \hline
 0 \phantom{1} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0} \phantom{0}
 \end{array}$$

$11101_2 * 1110_2 = 110010110_2$

**Interi positivi e negativi**

Torniamo sulla sottrazione; non è un caso che tutti gli esempi proposti sono caratterizzati da un valore del minuendo maggiore del valore del sottraendo, poiché finora abbiamo considerato la codifica e, quindi le operazioni, tra numeri naturali (numeri interi senza segno). Il passo successivo è quello esaminare la codifica dei numeri con segno, cioè dei numeri interi (positivi, negativi e 0).

Nota bene: le argomentazioni successive hanno valore solo se si prendono in esame valori numerici codificati con un numero prefissato di cifre.

Tra i diversi metodi per rappresentare i numeri interi in base 2, prendiamo in considerazione

- la rappresentazione in modulo e segno (M&S)
- la rappresentazione in complemento a 2

### Rappresentazione in modulo e segno

Dopo aver stabilito il numero N di bit che si vogliono utilizzare per la codifica, si riserva il bit più significativo per il segno, tutti gli altri (N-1) sono utilizzati per codificare il valore assoluto del numero in esame.

La convenzione adottata è di utilizzare il valore 1 per il segno negativo, 0 per quello positivo.

Esempio: con N=4 bit

numero = -3 → 1011 dove 1 rappresenta il segno -

numero = +3 → 0011 dove 0 rappresenta il segno +

→ i rimanenti 3 bit (011) rappresentano il valore assoluto, ovviamente uguale in entrambe le codifiche

Osservazioni:

- lo 0 ha una doppia rappresentazione, una positiva e una negativa;
- l'intervallo di valori rappresentabili con N bit è  $[-2^{N-1}+1, +2^{N-1}-1]$ .

Alla facilità di codifica corrispondono però problemi di elaborazione: il fatto che il bit più significativo (il bit del segno) ha un significato diverso dagli altri implica che deve essere trattato in modo diverso nelle operazioni.

Esempio: Calcoliamo -20 +2 (supponiamo di utilizzare codifiche su 8 bit)

$$\begin{array}{r}
 -20 = 10010100 \quad \text{quindi} \quad 10010100 + \\
 + 2 = 00000010 \quad \underline{\quad\quad\quad 00000010} = \\
 \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 10010110
 \end{array}$$

che corrisponde a -22 (non a -18 che è il risultato corretto)

Il problema deriva dal fatto che per il bit del segno non vale più la regola del peso associato alla posizione.

Si preferisce utilizzare, invece, la notazione in complemento a 2, che ci permette di trasformare la sottrazione in addizione.

### Rappresentazione in complemento a 2

Definiamo il significato di complemento ad una base:

in generale si definisce complemento ad una base M di un numero X rappresentato con N cifre, quel numero Y che sommato a X dà come risultato la potenza N-esima della base.

Difficile? No, se rileggiamo quanto detto attraverso degli esempi:

esempio 1:	se X= 6, M= 10 e N= 1, la potenza $10^1$ è 10; quanto vale Y, ovvero quale è quel numero che sommato a 6 dà come risultato 10?
	la risposta, ovvia, è Y= 4
esempio 2:	se X= 38, M= 10 e N= 2, la potenza $10^2$ è 100; quanto vale Y?
	la risposta è Y= 62 (62+38 = 100).

E se la base è 2? Il procedimento è il medesimo; esaminiamolo attraverso un esempio:

se X= 11001 e N= 5, la potenza  $2^5$  è 100000, quindi il numero che sommato a X dà come risultato 100000 è Y=00111.

Infatti

$$\begin{array}{r}
 11001 + \\
 \quad\quad 00111 = \\
 100000
 \end{array}$$

Leggiamo quanto svolto in base 10:  $11001_2 = 25$ , la potenza  $2^5$  è  $32 = 100000$  (ricordati che devi ragionare in base 2).

Quale è quel numero che sommato a 25 dà 32?

Risposta: 7 che in base 2 è proprio 111.

Introduciamo ora la rappresentazione dei numeri interi in complemento a 2 (con N bit).

Definizione:

- la rappresentazione di un numero positivo coincide con la rappresentazione in modulo e segno (entrambe con N bit)
- la rappresentazione di un numero negativo è costituita dal complemento a 2 del corrispondente numero positivo.

Osservazioni:

- l'intervallo di valori rappresentabili con N bit è  $[-2^{N-1}, +2^{N-1}-1]$ , quindi è possibile rappresentare un valore in più rispetto alla rappresentazione in modulo e segno.
- i numeri positivi iniziano tutti con un bit a 0
- i numeri negativi iniziano tutti con un bit a 1
- esiste una sola rappresentazione dello 0: (0 ... 0).

Riportiamo nella successiva tabella la codifica in modulo e segno e in complemento a 2 dei numeri interi rappresentabile con N=4 bit.

in base 10	modulo e segno	complemento a 2
-8	-----	1000
-7	1111	1001
-6	1110	1010
-5	1101	1011
-4	1100	1100
-3	1011	1101
-2	1010	1110
-1	1001	1111
-0	1000	-----
+0	0000	0000
+1	0001	0001
+2	0010	0010
+3	0011	0011
+4	0100	0100
+5	0101	0101
+6	0110	0110
+7	0111	0111

Esiste una regola di facile applicazione che permette di scrivere la rappresentazione di un numero negativo in complemento a 2 con N bit:

1. si scrive la codifica con N bit del numero positivo corrispondente
2. si lascia inalterata, partendo dal bit meno significativo, la sequenza di bit fino al primo 1 che si incontra, che rimane anch'esso invariato; si modificano tutti i altri bit: 0 diventa 1, 1 diventa 0.

Esempio: vogliamo codificare in base 2 con 6 bit il corrispondente valore di -22 in base 10																			
1° passo:	+ 22 <sub>10</sub> = 010110 <sub>2</sub>																		
2° passo:	si complementa bit a bit, applicando la regola appena esposta																		
	<table style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td> </tr> <tr> <td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td><td>↓</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td> </tr> </table>	0	1	0	1	1	0	↓	↓	↓	↓	↓	↓	1	0	1	0	1	0
0	1	0	1	1	0														
↓	↓	↓	↓	↓	↓														
1	0	1	0	1	0														
	quindi -22 <sub>10</sub> =101010 <sub>2</sub> in complemento a 2																		

Affrontiamo ora il problema posto all'inizio del paragrafo:

la sottrazione tra numeri relativi.

Partiamo dalla considerazione che la differenza tra due numeri equivale alla somma algebrica tra il primo operando e l'opposto del secondo.

Infatti		$  \begin{array}{r}  23 - \\  \underline{12} = \rightarrow \text{equivale} \\  +11  \end{array}  $	$23 + (-12) = +11$
È sufficiente rappresentare il sottraendo in complemento a 2, trasformando così una sottrazione in un'addizione.			

Enunciamo la regola per eseguire la somma algebrica tra 2 numeri interi codificati in complemento a 2 su N bit:

- si esprimono i numeri in complemento a 2 su N bit (la codifica differisce solo per i numeri negativi)
- si esegue la somma
- si trascura l'eventuale overflow (ovvero il bit di posizione N+1)

Applichiamo la regola (negli esempi successivi i numeri sono codificati su 6 bit)

Esempio:	Operandi di segno opposto		
	in questo caso il valore ottenuto è sempre corretto (il numero di bit è sufficiente per memorizzare il risultato)		
	-22 = 101010	101010 +	
	+ 2 = 000010	<u>000010</u> =	
		101100	
	Nel risultato il bit del segno è 1 → valore negativo, quindi occorre complementare 101100 diventa 010100 = 20		
		risultato finale [ -22 + 2 = -20 ] → risultato corretto	
Esempio:	Operandi di segno concorde +		<b>è sempre senza overflow</b> (perché?)
	in questo caso il valore ottenuto non sempre è corretto		
	risultato corretto (senza overflow)	risultato errato (senza overflow)	
	+17 + 8 = +25	+17 +16 = +33	
	+17 = 010001	010001 +	010001 +
	+8 = 001000	<u>001000</u> =	<u>010000</u> =
		011001	100001
	poiché [011001] <sub>2</sub> = 25 <sub>10</sub> ]		Nel risultato il bit del segno è 1 → valore negativo
	risultato corretto		risultato errato

Esempio: Operandi di segno concorde - **è sempre con overflow** (perché?)

in questo caso il valore ottenuto non sempre è corretto

risultato corretto (con overflow)		risultato errato (con overflow)	
-24 + (-7) = -31		-24 + (-12) = -36	
-24 = 101000	101000 +	-24 = 101000	101000 +
-7 = 111001	<u>110100</u> =	-12 = 110100	<u>110100</u> =
	1100001		1011100

Il bit più significativo è il bit di overflow → si trascura (è il 7° bit), il bit immediatamente successivo è il bit del segno: valore 1 quindi il risultato si complementa

Il bit più significativo è il bit di overflow → si trascura (è il 7° bit), il bit immediatamente successivo è il bit del segno: valore 0 quindi il risultato è positivo

poiché [1100001]<sub>2</sub> = -31<sub>10</sub>]

risultato corretto

risultato errato

Esercizi: eseguire le seguenti operazioni in aritmetica binaria, codificare i numeri in decimale e controllare i risultati ottenuti

**addizione**

- |                             |                      |
|-----------------------------|----------------------|
| 1) $111_2 + 1_2$            | [8 <sub>10</sub> ]   |
| 2) $1111_2 + 1101_2$        | [28 <sub>10</sub> ]  |
| 3) $1001001_2 + 1110_2$     | [87 <sub>10</sub> ]  |
| 4) $1111101_2 + 11111111_2$ | [380 <sub>10</sub> ] |
| 5) $1000000_2 + 10000000_2$ | [192 <sub>10</sub> ] |
| 6) $1100100_2 + 11011_2$    | [127 <sub>10</sub> ] |
| 7) $1000000_2 + 11111_2$    | [95 <sub>10</sub> ]  |

**sottrazione (tra numeri naturali)**

- |                             |                      |
|-----------------------------|----------------------|
| 8) $100000_2 - 10000_2$     | [16 <sub>10</sub> ]  |
| 9) $100110_2 - 110_2$       | [32 <sub>10</sub> ]  |
| 10) $1111000_2 - 10100_2$   | [100 <sub>10</sub> ] |
| 11) $1001000_2 - 1000001_2$ | [7 <sub>10</sub> ]   |
| 12) $1000110_2 - 110010_2$  | [20 <sub>10</sub> ]  |
| 13) $1011011_2 - 100110_2$  | [53 <sub>10</sub> ]  |
| 14) $1011000_2 - 11010_2$   | [62 <sub>10</sub> ]  |

**moltiplicazione**

- |                         |                      |
|-------------------------|----------------------|
| 15) $101_2 * 10_2$      | [10 <sub>10</sub> ]  |
| 16) $101_2 * 110_2$     | [30 <sub>10</sub> ]  |
| 17) $1010_2 * 111_2$    | [70 <sub>10</sub> ]  |
| 18) $11011_2 * 1100_2$  | [324 <sub>10</sub> ] |
| 19) $11111_2 * 111_2$   | [217 <sub>10</sub> ] |
| 20) $11111_2 * 1111_2$  | [465 <sub>10</sub> ] |
| 21) $11111_2 * 11111_2$ | [961 <sub>10</sub> ] |

**divisione**

- |                             |   |
|-----------------------------|---|
| 22) $1100_2 : 11_2$         | [Q= 4 <sub>10</sub> ; R= 0 <sub>10</sub> ]  |
| 23) $101111_2 : 101_2$      | [Q= 9 <sub>10</sub> ; R= 2 <sub>10</sub> ]  |
| 24) $101111_2 : 111_2$      | [Q= 6 <sub>10</sub> ; R= 5 <sub>10</sub> ]  |
| 25) $1010000_2 : 1011_2$    | [Q=7 <sub>10</sub> ; R= 3 <sub>10</sub> ]   |
| 26) $1111101_2 : 110_2$     | [Q= 20 <sub>10</sub> ; R= 5 <sub>10</sub> ] |
| 27) $10001000_2 : 1111_2$   | [Q= 9 <sub>10</sub> ; R= 1 <sub>10</sub> ]  |
| 28) $100000000_2 : 10000_2$ | [Q= 16 <sub>10</sub> ; R= 0 <sub>10</sub> ] |

**somma algebrica (in complemento a 2 su 7 bit)**

- |                             |                            |
|-----------------------------|----------------------------|
| 29) $0100011_2 + 0101101_2$ | [-48 <sub>10</sub> errore] |
| 30) $0100011_2 + 0001100_2$ | [47 <sub>10</sub> ]        |
| 31) $1010011_2 + 0010111_2$ | [-22 <sub>10</sub> ]       |
| 32) $0110011_2 + 1000100_2$ | [-9 <sub>10</sub> ]        |
| 33) $0011011_2 + 1100100_2$ | [-1 <sub>10</sub> ]        |
| 34) $1000100_2 + 1001110_2$ | [-46 <sub>10</sub> errore] |
| 35) $1110001_2 + 1001111_2$ | [-64 <sub>10</sub> ]       |
| 36) $1101110_2 + 1110010_2$ | [-32 <sub>10</sub> ]       |





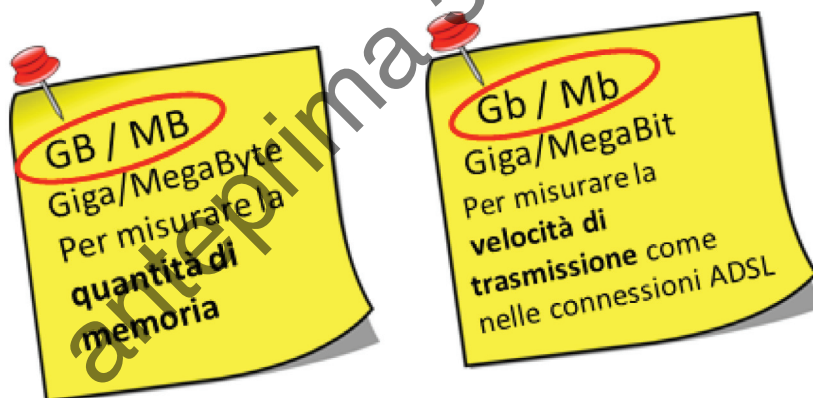
## Codifica dei caratteri

I numeri rappresentano solo una piccola parte delle informazioni memorizzate ed elaborate da un computer; insieme alla codifica delle informazioni numeriche è sorto il problema della rappresentazione dei dati alfanumerici, ovvero dei documenti testuali e anche in questo caso sono state elaborate leggi di trasformazione da carattere a stringa di bit. Prima però di entrare nel dettaglio è fondamentale introdurre la definizione di byte.

Si definisce BYTE una sequenza di 8 BIT (evidente anche in questo caso la naturale predilezione che si ha in informatica per le potenze del 2 come conseguenza dell'uso della codifica binaria). Essendo codificabili su 8 bit 256 combinazioni distinte di 0/1, si ritenne che tale quantità di bit fosse sufficiente per rappresentare tutti i simboli utilizzabili nella scrittura di messaggi e documenti e da allora il BYTE è divenuta una sorta di unità di misura della codifica (1 BYTE = 1 carattere), tuttora utilizzata per definire la capacità di memoria: sempre privilegiando le potenze del 2 e utilizzando il simbolo "B" per riferirsi al byte (ricorda che "b" è usato per indicare grandezze espresse in bit), si definisce:

sigla	descrizione	Quantità in byte
1kB	1 kilo-byte	1024 bytes
1MB	1 mega-byte	$1024^2$ bytes = 1.048.576 bytes
1GB	1 giga-byte	$1024^3$ bytes = 1.073.741.824 bytes
1TB	1 tera-byte	$1024^4$ bytes = 1.099.511.627.776 bytes

Una unità a disco da 1TB (un tera-byte) è dunque in grado di memorizzare più di mille miliardi di caratteri.



Si è detto che su 1 byte cioè su 8 bit è possibile codificare un carattere: come stabilire allora la corrispondenza tra byte e carattere? La corrispondenza non può che essere basata su una "convenzione", cioè su un accordo stabilito da un'apposita commissione composta da tecnici, studiosi, rappresentanti di aziende, istituzioni e governi che scelgono uno standard sul quale uniformarsi, i due standard più diffusi e noti sono il codice ASCII e il codice UNICODE.

Il codice **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange ovvero *codifica standard americana per lo scambio di informazioni*) è nato come codice a 7 bit: sono sufficienti infatti 7 bit (quindi 128 combinazioni di 0/1) per rappresentare tutti i caratteri maiuscoli e minuscoli dell'alfabeto inglese, le cifre da 0 a 9, i simboli di punteggiatura, le parentesi, gli operatori matematici ("+", "-", "\*", "\", "%"), gli operatori relazionali ("=", ">", "<") oltre ad alcuni caratteri "non stampabili", cioè non corrispondenti a simboli, ma testuali utilizzati originariamente come comandi nelle trasmissioni telegrafiche. All'inizio degli anni 1960 con l'affermarsi del byte anche come unità di misura delle memorie, fu aggiunto un bit, passando così dalla codifica ASCII alla codifica ASCII estesa; l'aggiunta di un bit ha permesso di raddoppiare il numero di caratteri codificabili (da 128 a 256), permettendo la codifica dei caratteri nazionali (per esempio le vocali accentate: ora è possibile codificare *età* e non *eta'*).

L'estendersi delle comunicazioni tra paesi che utilizzano alfabeti diversi dall'alfabeto inglese ha comportato la necessità di ampliare ulteriormente l'insieme dei simboli da codificare e, conseguentemente, è stato giudicato inadeguato un solo byte per carattere; da qui il diffondersi del codice UNICODE, che associa ad ogni carattere 16 bit, cioè 2 byte e che quindi permette la codifica di  $2^{16} = 65536$  simboli diversi.

Segue la tabella della codifica ASCII standard su 8 bit (i primi 128 caratteri)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr	Dec	Hx	Oct	Chr
0	0	000	NUL (null)	32	20	040	Space	64	40	100	@	96	60	140	`
1	1	001	SOH (start of heading)	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX (start of text)	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX (end of text)	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT (end of transmission)	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ (enquiry)	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK (acknowledge)	38	26	046	^	70	46	106	F	102	66	146	f
7	7	007	BEL (bell)	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS (backspace)	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB (horizontal tab)	41	29	051	)	73	49	111	I	105	69	151	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT (vertical tab)	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR (carriage return)	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO (shift out)	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI (shift in)	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE (data link escape)	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1 (device control 1)	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2 (device control 2)	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3 (device control 3)	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4 (device control 4)	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN (synchronous idle)	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB (end of trans. block)	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN (cancel)	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM (end of medium)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB (substitute)	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC (escape)	59	3B	073	;	91	5B	133	[	123	7B	173	{
28	1C	034	FS (file separator)	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS (group separator)	61	3D	075	=	93	5D	135	]	125	7D	175	}
30	1E	036	RS (record separator)	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US (unit separator)	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Riportiamo per completezza anche i successivi 128 caratteri della codifica ASCII estesa (solo la codifica decimale)

128	Ç	144	É	160	á	176	ÿ	192	ÿ	209	ƒ	225	ß	241	±
129	ü	145	æ	161	í	177	ÿ	193	ÿ	210	ƒ	226	Γ	242	≥
130	é	146	Æ	162	ó	178	ÿ	194	ÿ	211	ƒ	227	π	243	≤
131	â	147	ø	163	ú	179	ÿ	195	ÿ	212	ƒ	228	Σ	244	ƒ
132	ä	148	ö	164	ñ	180	ÿ	196	ÿ	213	ƒ	229	σ	245	ƒ
133	à	149	ò	165	ñ	181	ÿ	197	ÿ	214	ƒ	230	μ	246	+
134	â	150	û	166	ÿ	182	ÿ	198	ÿ	215	ÿ	231	τ	247	≈
135	ç	151	ù	167	ÿ	183	ÿ	199	ÿ	216	ÿ	232	Φ	248	°
136	ê	152	ÿ	168	ÿ	184	ÿ	200	ÿ	217	ÿ	233	⊙	249	·
137	ë	153	ÿ	169	ÿ	185	ÿ	201	ÿ	218	ÿ	234	Ω	250	·
138	è	154	ÿ	170	ÿ	186	ÿ	202	ÿ	219	ÿ	235	δ	251	√
139	ï	156	ÿ	171	½	187	ÿ	203	ÿ	220	ÿ	236	∞	252	—
140	î	157	ÿ	172	¼	188	ÿ	204	ÿ	221	ÿ	237	φ	253	²
141	ì	158	ÿ	173	ÿ	189	ÿ	205	ÿ	222	ÿ	238	e	254	■
142	Ä	159	f	174	«	190	ÿ	206	ÿ	223	ÿ	239	∧	255	
143	Å	192	L	175	»	191	ÿ	207	ÿ	224	α	240	≡		

A titolo di esempio, la sequenza ASCII di codifica del testo "Ciao, mondo!":

C	i	a	o	,		m	o	n	d	o	!
in esadecimale											
43	69	61	6F	2C	20	6D	6F	6E	64	6F	21
in binario											
01000011	01101001	01100001	01101111	00101100	00010000	01101101	01101111	01101110	01100100	01101111	00100001

E' piuttosto evidente che, se dovessimo codificare in binario un documento senza l'aiuto di un qualche automatismo, ci troveremmo in grave difficoltà così come nella stessa difficoltà si troverebbe chi dovesse decodificarlo nella necessità di ricostruire il documento originale

"Ciao, mondo!" > codificatore > sequenza di bit  
sequenza di bit > decodificatore > "Ciao, mondo!"

Nel precedente schema il "codificatore" e il "decodificatore" sono degli strumenti che, conoscendo perfettamente il criterio (cioè le regole) di codifica, sono in grado di effettuare tanto la trasformazione da documento testuale a documento binario quanto la trasformazione inversa.

## Codifica delle immagini (fisse)

Così come un testo può essere memorizzato su un supporto elettronico, dopo essere stato codificato in una sequenza di bit grazie a codici come ASCII o UNICODE, anche le immagini, i suoni e i video possono essere memorizzati purché opportunamente “trasformati” in sequenze di 0 e 1. A differenza, però, di un documento testuale, un'immagine non nasce come un oggetto discreto, ovvero costituito da un insieme finito e ben determinato di simboli (caratteri dell'alfabeto inglese maiuscoli e minuscoli, cifre, simboli aritmetici, segni di punteggiatura, etc.), ma è un insieme di colori, linee, forme, sfumature che combinate tra loro rappresentano appunto un'immagine (una fotografia è un'immagine, un quadro è un'immagine, la stampa di un carattere è un'immagine). Occorre quindi trasformare l'immagine in una opportuna sequenza di simboli che, a loro volta, verranno codificati in sequenze di 0 e 1. Più precisamente occorre DIGITALIZZARE l'immagine, cioè passare dalla sua rappresentazione analogica alla corrispondente rappresentazione digitale.

Ma cosa intendiamo in generale con i termini analogico e digitale?

Una grandezza si definisce **analogica** quando varia in modo continuo nel tempo e/o nello spazio e nei valori assunti, cioè quando può assumere qualsiasi valore in un determinato intervallo. Una grandezza si definisce **digitale** quando NON varia in modo continuo: essa può assumere valori che sono elementi di un insieme discreto e quindi può assumere in un determinato intervallo solo alcuni valori e non altri (si dice che una grandezza digitale assume valori in modo discontinuo o a salti).

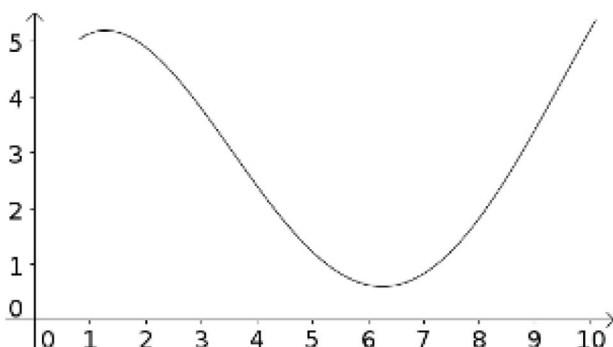
Consideriamo due grandezze fisiche con cui interagiamo quotidianamente, tempo e temperatura: siamo abituati a rappresentare il loro valore sia in “analogico” che in “digitale”.

Grandezza (e relativa unità di misura)	Misura analogica	Misura digitale
Tempo (ora, minuti)		
temperatura (gradi)		

Torniamo al termine digitalizzare con una definizione: “Nel campo dell'informatica e dell'elettronica, con digitalizzazione si intende il processo di trasformazione di un'immagine, di un suono, di un documento in un formato digitale, interpretabile da un computer, dove per formato digitale si intende un codice binario in cui tutto è rappresentato da combinazioni di zero od uno, quindi da stati del tipo acceso/spento.” (Wikipedia)

il processo di digitalizzazione è composto da una sequenza di tre fasi:

1. il campionamento.
2. la quantizzazione.
3. la codifica.

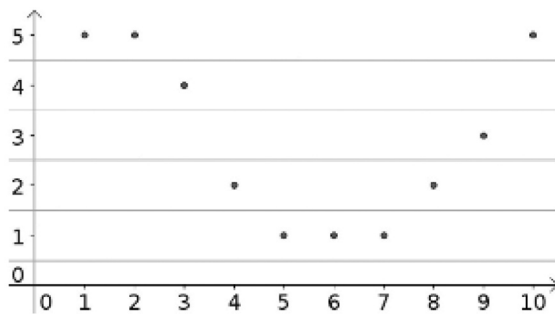


Considerare un segnale analogico qualsiasi: esso può essere rappresentato graficamente tramite una “linea” come nella figura a fianco.

Possiamo, ad esempio, pensare che sull'asse delle ascisse rappresentiamo il tempo e sull'asse delle ordinate il valore che il segnale assume in quell'istante. I valori cambiano con continuità (la funzione è continua) nell'intervallo di tempo considerato.

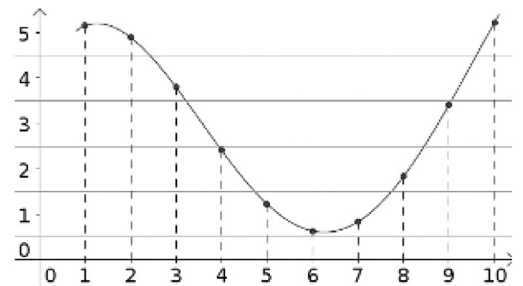
Eseguiamo i passi necessari per “digitalizzare” questa grandezza.

Campionamento: i valori della grandezza vengono rilevati ad intervalli di tempo distinti (la grandezza è ancora analogica).



Codifica: ad ogni valore che la grandezza assume nell'intervallo di campionamento viene associato un numero (una sequenza di 0 e 1) in base ad un codice.

Dopo questa digressione, torniamo all'argomento del paragrafo, la digitalizzazione delle immagini: descriveremo due strategie, la grafica bitmap o raster e la grafica vettoriale.



Quantizzazione: si definisce l'intervallo di valori all'interno del quale ogni singolo valore campionato viene approssimato (in questa fase si introduce un errore che è inversamente proporzionale al numero di valori che vengono fissati).

Campione	Valore	CODIFICA			
1	5	0	1	0	1
2	5	0	1	0	1
3	4	0	1	0	0
4	2	0	0	1	0
5	1	0	0	0	1
6	1	0	0	0	1
7	1	0	0	0	1
8	2	0	0	1	0
9	3	0	0	1	1
10	5	0	1	0	1

**Grafica bitmap o raster.**

Consiste nel sovrapporre all'immagine una griglia di punti o, meglio, di suddividere l'immagine in una serie di celle di dimensioni ridotte (quanto ridotte lo vedremo tra poco): ogni punto o cella prende il nome di pixel (PICture ELeMent); il pixel è quindi il più piccolo elemento costituente un'immagine digitalizzata. Ma in quanti pixel possiamo suddividere un'immagine? Non esiste un valore fisso, esiste però il concetto di “fedeltà” all'immagine iniziale o, meglio, di “qualità” dell'immagine riprodotta. Vediamo di spiegare questo concetto considerando, come esempio, le seguenti immagini:

		
Larghezza 25 pixel (*) Altezza 27 pixel	Larghezza 64 pixel (*) Altezza 67 pixel	Larghezza 256 pixel Altezza 270 pixel

(\*) valori approssimati  
(N.B. tutte e tre le immagini hanno la stessa dimensione)


Possiamo quindi affermare che all'aumentare del numero di pixel, mantenendo costanti le dimensioni della figura, riusciamo ad ottenere un'immagine “migliore” cioè sempre più fedele all'originale.

Il numero di pixel in cui si suddivide lo schermo prende il nome di **definizione**, a cui è strettamente connesso, ma non è un sinonimo, il concetto di **risoluzione**, che coincide con la quantità di pixel per unità di misura. La risoluzione è quindi la densità di pixel e da essa dipende la qualità dell'immagine; nell'esempio in figura abbiamo assunto come unità di misura la dimensione dell'immagine e modificato la quantità di pixel in essa contenuti: aumentando la densità è aumentata la qualità. L'unità di misura più frequentemente usata per quantificare la risoluzione è il **dpi** (Dot Per Inches), numero di pixel per pollice (un pollice= 2,54 cm).

Esiste un terzo parametro che caratterizza un'immagine digitalizzata: la **profondità** dell'immagine (o del colore), anch'essa rappresentata con un certo numero di bit: maggiore è il numero di bit utilizzati maggiore è la qualità del colore riprodotto.

Infatti un'immagine è fatta anche e soprattutto di colori, quindi ad un pixel occorre associare un numero che codifica il colore che ha, nell'immagine da riprodurre, il punto rappresentato.



Per semplicità supponiamo di voler rappresentare un carattere, per esempio il carattere 'R', in bianco e nero: basterà un solo bit per rappresentare il colore, per esempio 1 per il nero, 0 per il bianco.

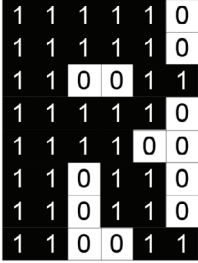
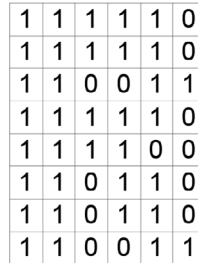

<b>Immagine del carattere 'R'</b>	<b>Immagine con sovrapposta una griglia di 4x3 pixel</b>
	

(le due figure successive sono state ingrandite per permettere una maggiore leggibilità)

<b>Associamo ad ogni pixel un bit in base al colore <u>predominante</u> nel quadratino</b>	<b>Togliendo il colore, rimane la codifica che letta come sequenza di bit è 111101110101</b>	<b>Facendo il procedimento inverso, sostituiamo ai bit il colore (sempre con la stessa regola: 0 → bianco, 1 → nero) e otteniamo</b>
		 che non coincide esattamente con l'immagine iniziale.

Proviamo a ridurre la grandezza di un pixel ovvero ad aumentare il numero di quadratini della griglia da sovrapporre all'immagine.



<b>Immagine del carattere 'R'</b>	<b>Immagine con sovrapposta una griglia di 8x6 pixel</b>
	

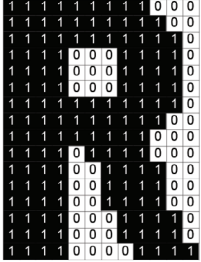
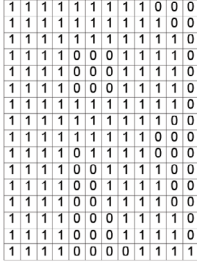

<b>Ripetiamo gli stessi passi ...</b>	<b>Otteniamo la sequenza 111110 111110 110011 111110 111100 110110 110110 110011</b>	<b>che nuovamente trasformata diventa</b>
		

più fedele all'originale anche se con una "particolarità". Domanda: sai spiegarne il perché?



Un ultimo raffinamento.

Immagine del carattere 'R'	Immagine con sovrapposta una griglia di 12x16 pixel
	

Ripetiamo gli stessi passi ...	Quale sequenza di bit otteniamo?	E infine ...
		

Non coincide certamente con l'immagine iniziale ma è "più simile" ad essa rispetto alle precedenti.

La codifica dei soli colori nero e bianco è accettabile se l'immagine è un carattere, non lo è se l'immagine è una fotografia, un disegno, un dipinto. Infatti anche una fotografia in bianco e nero di fatto non contiene solo questi 2 colori ma diverse tonalità di grigio (per le sfumature, i contrasti, le ombre e quant'altro). Immaginiamo allora di utilizzare 4 bit per codificare il colore di un singolo pixel: significa avere a disposizione 16 ( $=2^4$ ) diverse combinazioni di 0 e 1, ognuna delle quali può essere associata ad un pixel e, quindi, ogni pixel può assumere un colore che varia tra i 16 possibili. Più precisamente non si parla di 16 colori ma di 16 tonalità di grigio che vanno da 0000 (bianco) a 1111 (nero) passando per varie sfumature di grigio (alla codifica 0001 corrisponderà un grigio chiarissimo, alla codifica 1110 un grigio molto scuro).

A questo punto è chiaro che, aumentando il numero di bit impiegati per la codifica del colore, aumenta la quantità di toni di grigio che possono essere rappresentati: con 8 bit è possibile rappresentare 256 ( $=2^8$ ) toni di grigio.

Ma, se vogliamo codificare un'immagine con i suoi colori, occorre approfondire alcuni concetti riguardanti la teoria del colore.

### La rappresentazione RGB del colore

La "teoria del colore" afferma che una parte piuttosto ampia dei colori percepiti dai nostri occhi può essere immaginata come sovrapposizione di TRE componenti monocromatiche dette PRIMARIE; il lettore avrà certamente avuto modo di sperimentare come, nella pittura, alcuni particolari tipi di ROSSO, BLU e GIALLO si prestino ad essere combinati tra loro per riprodurre gli altri colori: il BIANCO viene usato come una sorta di diluente cioè di dispersore del pigmento colorato con l'obiettivo di ottenere tonalità più chiare, mentre il NERO è utilizzato per scurire i toni; la necessità del "nero" (a dispetto della teoria) scaturisce dal fatto che il miscuglio in parti uguali dei tre colori primari produce il cosiddetto BISTRO, un nero piuttosto imperfetto a causa delle impurità presenti nei colori primari. Anche nella tecnica tipografica si fa uso dello stesso principio adottando il MAGENTA (un rosso violaceo), il CIANO (un celeste) e il GIALLO come colori primari con l'aggiunta del NERO necessario per aggiungere "forza" alle tonalità scure ottenute con la sovrapposizione dei tre colori primari; nella tecnica tipografica si parla di "quadricromia" e si usa l'acronimo "CMYK" intendendo proprio "C" per "cyan" (ciano), "M" per "magenta", "Y" per "yellow" (giallo) e infine "K" che, nonostante in origine indicasse altro ("Key plate", tecnica di allineamento delle lastre CMY), nella prassi corrente si intende per "nero".






Tanto in pittura che in tecnica tipografica, tuttavia, il colore viene riprodotto per SINTESI SOTTRATTIVA in quanto ciò che i nostri occhi vedono e percepiscono come colore è la parte riflessa della luce, incidente su ciò che osserviamo: la parte non riflessa viene in effetti assorbita, dunque SOTTRATTA, dal pigmento colorato.

In effetti siamo in grado di “vedere” un quadro oppure un'immagine stampata SOLO se siamo in un ambiente luminoso: riuscireste a vedere qualcosa in una stanza perfettamente al buio?




Ben diversa è la situazione quando guardiamo lo schermo del computer oppure un televisore: a differenza di quadri e stampe lo schermo EMETTE la luce. Si parla allora di riproduzione del colore per SINTESI ADDITIVA in quanto ciò che vediamo in effetti è la sovrapposizione di tre sorgenti luminose primarie che in questo caso sono state individuate nella luce ROSSA, nella luce VERDE e nella luce BLU unite nella famosa sigla RGB (“Red”, “Green”, “Blue”).

Nella codifica RGB per ogni pixel occorre distinguere tre distinti valori riferibili, nell'ordine, alle componenti di luci rossa, verde e blu; utilizzando un byte per ogni colore primario ogni pixel è codificato con 3 byte (3\*8=24 bit in tutto). Ogni gruppo di 8 bit è relativo ad un colore primario e può essere pensato come la quantità di quel colore nell'insieme; i primi 8 bit rappresentano quindi la quantità di rosso che c'è in quel pixel, i successivi 8 la quantità di verde e gli ultimi 8 la quantità di blu. Oltre che con la codifica binaria, ogni pixel può essere rappresentato con la corrispondente codifica decimale o esadecimale (in entrambe le codifiche vengono utilizzati 3 numeri, uno per ciascuna delle 3 sequenze di 8 bit), migliorando così la leggibilità della rappresentazione.

Facciamo qualche esempio:

COLORE	BINARIO			DECIMALE			ESADECIMALE		
	R	G	B	R	G	B	R	G	B
rosso 	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	255	0	0	FF	00	00
verde 	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0	255	0	00	FF	00
blu 	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1	0	0	255	00	00	FF

Osserviamo che il giallo, colore primario percepito dall'occhio umano, non è presente nella codifica RGB, ma è rappresentabile come combinazione di rosso e verde; facciamo qualche esempio, iniziando con la codifica di una tonalità più luminosa di giallo, passando per una tonalità più scura e finire con una tonalità più chiara:


COLORE (giallo)	BINARIO			DECIMALE			ESADECIMALE		
	R	G	B	R	G	B	R	G	B
luminoso 	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	255	255	0	FF	FF	00
scuro 	1 1 1 0 0 1 1 0	1 1 1 0 0 1 1 0	0 0 1 0 0 1 0 0	230	230	102	E6	E6	4C
chiaro 	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 1 1 0 0 1 1 0	255	255	102	FF	FF	66

Utilizzando 3 byte, quindi 24 bit, possiamo codificare ben  $2^{24} = 16.777.216$  colori distinti: un'enormità, tanto da ipotizzare, nel criterio di codifica, l'uso di un numero minore di bit senza un sensibile degrado della percezione di un'immagine ma con il vantaggio di un minore “peso” in termini di quantità di byte.

Un altro modo per contenere il numero di byte necessari per memorizzare un'immagine a colori è quello di utilizzare un sottoinsieme dei 16.777.216 colori distinti. I colori selezionati vengono memorizzati in una "tavolozza" di colori o "palette", ogni colore è codificato con lo standard RGB e occupa una posizione nella tavolozza; ad ogni pixel è associato un numero che non rappresenta più il colore del pixel nell'immagine ma la posizione che ha il colore del pixel in esame nella tavolozza del colore. Ovviamente oltre alla stessa immagine memorizzata bisognerà anche salvare la palette, una sorta di codice dei colori utilizzati che consentirà la giusta visualizzazione.

Ad esempio prendiamo in considerazione un'immagine composta da 12 pixel (per comodità li abbiamo numerati da 1 a 12) in cui sono presenti solo 8 diversi colori.

Costruiamo la tavolozza che conterrà gli 8 colori che vengono utilizzati nell'immagine: ad ogni pixel è associato una stringa di bit che rappresenta la posizione che il colore del pixel in esame ha nella palette.



Esempio: il pixel 1 è di colore rosso, il rosso occupa nella palette la posizione 101, di conseguenza al pixel 1 verrà associata la codifica 101.

Codifica con PALETTE

pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel
1	2	3	4	5	6	7	8	9	10	11	12
101	000	100	001	110	011	010	111	101	110	011	001

000	13	F3	45
001	90	68	E5
010	AB	10	80
011	CD	E6	F6
100	F6	BC	E6
101	FF	00	00
110	FF	FF	33
111	FF	AA	55

(codice HEX RGB)

Calcoliamo quanta memoria occorre per l'immagine : 12 (pixel) x 3 bit = 36 bit  
 per la palette: 8 x 24 bit = 192 bit  
 dove: 8 = numero di elementi E 24 = numero di bit per la codifica in RGB  
 totale: **228 bit (= 36+192)**

In RGB abbiamo la seguente codifica

pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel	pixel
1	2	3	4	5	6	7	8	9	10	11	12
FF0000	13F345	F6BCE6	9068E5	FFFF33	CDE6F6	AB1080	FFAA55	FF0000	FFFF33	CDE6F6	9068E5

Calcoliamo quanta memoria occorre per l'immagine (solo per l'immagine)  
**12 (pixel) x 24 bit(codifica RGB)= 288 bit**  
 Generalmente, con la tavolozza c'è un risparmio nell'occupazione di memoria.

Domanda: ma se il colore che vogliamo rappresentare non è presente nella tavolozza?  
 Possiamo scegliere tra due possibili soluzioni: selezioniamo la tonalità, presente nella tavolozza, più vicina a quella da rappresentare accettando un margine di errore (approssimazione) oppure cambiamo palette.

**Grafica vettoriale (cenni)**

La codifica delle immagini mediante la grafica bitmap è adatta ormai solo ad alcune tipologie di immagini, sostituita nelle memorizzazioni di fotografie da formati compressi come per esempio il JPG, inoltre non si adatta a memorizzare disegni tecnici, diagrammi, mappe; in questi casi si preferisce utilizzare la grafica vettoriale che si basa sulla considerazione che ogni immagine può essere vista come un insieme di oggetti, i più utilizzati sono i luoghi geometrici come i punti e le rette. Quindi, dopo aver trasformato l'immagine che si intende digitalizzare in un ben determinato insieme di oggetti, si memorizzano le coordinate di tali elementi; occorrerà poi un programma in grado di leggere tali dati e trasformarli in pixel perché l'immagine iniziale venga riprodotta.

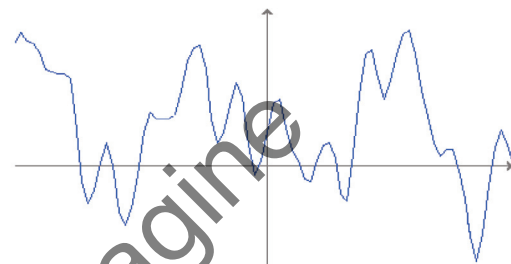
L'utilizzo della grafica vettoriale, oltre a comportare una minore occupazione di memoria rispetto alla memorizzazione della stessa immagine con la grafica bitmap, permette una maggiore manipolazione poiché è possibile modificare la dimensione dell'immagine (ingrandendo o rimpicciolendo) senza perdita di nitidezza; non è un caso che le mappe che è possibile consultare sul Web o sui navigatori GPS sono immagini vettoriali.

D'altra parte la codifica delle immagini con la grafica vettoriale dipende dal software utilizzato, di conseguenza non esiste uno standard universalmente riconosciuto.

Un'ultima nota: anche nel caso di grafica vettoriale c'è il problema di riprodurre il colore, ma la trattazione dell'argomento esula dagli obiettivi di questi appunti.

## Codifica dell'audio

Da un punto di vista fisico un suono può essere definito come una sensazione percepita dall'orecchio a causa del movimento di un corpo che mette in oscillazione l'aria. Tale movimento, detto più correttamente vibrazione, ha un andamento ondulatorio nel tempo e perciò un suono può essere rappresentato da un'onda, onda sonora.



Un suono è caratterizzato dalla *frequenza*, misurata in Hertz (Hz), che definisce l'altezza del suono, *dall'ampiezza* dell'onda, misurata in decibel (dB), che determina l'intensità del suono ed infine dalla *durata*.

La memorizzazione dei suoni su un disco di vinile può essere considerata una rappresentazione analogica del suono perché fornisce una rappresentazione continua dell'onda sonora, ma, ovviamente, non può essere memorizzata su un file ed elaborata da un computer.

E' necessario quindi convertire il segnale sonoro, analogico, in un segnale digitale; una tecnica utilizzata per la conversione analogico-digitale di un suono è quella di trasformare l'onda (il continuo) in un serie di valori rilevati in istanti di tempo distinti, di mantenere tali valori all'interno di un intervallo e codificare ciascun valore con un numero.

Ti ricorda qualcosa? Ebbene sì, è esattamente quanto già analizzato in dettaglio per la codifica delle immagini con la grafica bitmap; il processo di cui stiamo parlando è quello che, partendo dal campionamento del segnale e passando per la quantizzazione, arriva alla codifica dello stesso.

Non ripetiamo quanto già detto: con le dovute modifiche il procedimento è stato già descritto. Vogliamo solo ricordare alcuni dettagli:

- la griglia che si sovrappone al segnale digitale non ha carattere spaziale ma temporale;
- anche in questo caso il processo di quantizzazione introduce un errore come sempre avviene quando si passa da una grandezza continua ad un insieme discreto di numeri;
- maggiore è il numero di campioni, migliore è la qualità del suono riprodotto ma, ovviamente, maggiore è la quantità di memoria necessaria.

In ultimo, facciamo un po' di conti: nel processo di trasformazione il segnale originale viene trasformato in una sequenza di misure e, dunque, di numeri rappresentabili ovviamente in codifica binaria: 16 bit per ogni singolo campione (quindi 65.536 livelli distinti di ampiezza) sono solitamente sufficienti per una buona qualità di codifica del segnale acustico insieme ad una frequenza di campionamento di 44.100 Hz (cioè 44.100 campioni ogni secondo) utilizzata per una qualità CD (compact-disk).

Passiamo ai calcoli: quanti byte sono indispensabili per codificare quattro minuti di segnale audio (per esempio un brano musicale)? Presto detto:

Tempo (secondi)	N° totale di campioni (=campioni x tempo)	Occupazione (in byte e in Megabyte) (=n° totale campioni x byte occupati da un campione)
240 (=4 minuti * 60)	10.584.000 (=44.100*240)	21.168.000 (=10584000*2) = 20,18 MB (circa)

Quindi, un brano musicale di 4 minuti necessita di oltre 20 megabyte di codifica per una qualità audio di tipo CD (compact-disk).



## Codifica video

Alla base della codifica dei “filmati” c'è la riflessione, antica quanto la storia del cinema, che, osservando in rapida sequenza immagini fotografiche (dette “frame”) contenenti piccole variazioni, si ha la percezione della continuità e quindi del movimento. Una buona velocità in tal senso è 30 frame al secondo.

Se poi aggiungiamo una colonna sonora sincronizzata con le immagini otteniamo un video “completo” (suono+ immagine). Allo stesso modo, sequenze di immagini digitalizzate producono la stessa percezione: possiamo dunque pensare un video digitale come un oggetto codificato in un file contenente, in successione, i vari fotogrammi che compongono la sequenza video, a cui aggiungere i dati relativi alla digitalizzazione dei suoni (siano essi musica, voce umana, etc.) per completare il prodotto.

La digitalizzazione di un video è quindi una combinazione di tecniche di digitalizzazione di suoni e immagini; tra le inevitabili complicazioni insite in questo processo, vogliamo soffermare la nostra attenzione sul seguente quesito: quanto spazio occupa in termini di byte un filmato?

Facciamo alcune ipotesi e... qualche conto!

Immaginiamo che un filmato (senza sonoro per semplicità di calcolo) di 10 minuti sia composto da sequenze di immagini in alta definizione 1280x720 (per un totale di 921.600 pixel).

Ipotizziamo una codifica del colore RGB su 24 bit (3 byte): ogni frame è codificato su  $3 \times 921.600$  byte cioè 2.764.800 byte.

Se consideriamo 30 frame al secondo, equivalendo 10 minuti a 600 secondi, otteniamo un numero di frame complessivo pari a 18.000 e quindi 49.766.400.000 byte che valgono più di 46 GB (gigabyte): una quantità di memoria enorme per soli 10 minuti di filmato.

In generale, per calcolare l'occupazione di memoria di un file che contiene la digitalizzazione di un video occorre determinare quanto spazio occupa ogni unità elementare in cui è stato scomposto il video (immagini, suoni, caratteri), contare quante unità elementari di ciascun tipo sono presenti, calcolare l'occupazione totale di ciascun tipo e infine sommare i valori ottenuti.



## Classificazione dell'informazione

Giunti a questo punto, torniamo sulla prima definizione del capitolo che qui riportiamo arricchendola con l'aggiunta di un solo aggettivo:

“L'informatica si occupa della codifica BINARIA, della memorizzazione, della trasmissione e dell'elaborazione dell'informazione”.

Il riferimento all'uso generalizzato della rappresentazione binaria dell'informazione dovrebbe ora risultare chiaro: ogni genere di informazione trattata da sistemi automatici DEVE essere rappresentabile come sequenza di BIT.

Viceversa per poter riprodurre l'informazione codificata in una sequenza di bit è necessario conoscere i criteri adottati nella codifica. Una sequenza di bit del tipo:

```
101011110111011010001101011110111011011001101001100101010010000101001110100
101010111010000001101010110111111100110110000011010111111101110011111001
10001010101010101000001010101000101011110110101110010010111010100011101010101
```

non è interpretabile fintanto che non viene assegnato un criterio per decifrarla, ossia un criterio di decodifica. Ricordando la distinzione tra dato e informazione possiamo dire che

informazione = dato (ovvero sequenza di bit) + criterio di decodifica

Solo quando è stabilito il criterio di codifica (e dunque di decodifica) ha dunque senso parlare di digitalizzazione dell'informazione secondo il seguente schema:

INFORMAZIONE > CODIFICA > INFORMAZIONE CODIFICATA  
INFORMAZIONE CODIFICATA > DECODIFICA > INFORMAZIONE.

Un'ultima nota: l'informazione codificata viene memorizzata in un "oggetto" chiamato file a cui si assegna un NOME allo scopo di caratterizzare in modo chiaro ciò che rappresenta; nel caso per esempio di un documento di testo contenente il curriculum vitae di un tale "Mario", codificato in ASCII, si può assegnare un nome del tipo:

*curriculum\_mario.txt*

dove si distinguono due parti, la prima "curriculum\_mario" detta propriamente NOME; la seconda "txt" si dice ESTENSIONE e segue il PUNTO che fa da SEPARATORE.

In ambiente WINDOWS l'estensione indica il TIPO di informazione (quindi il criterio utilizzato nella codifica) che con una convenzione piuttosto diffusa nel mondo dell'Informatica indica un contenuto di tipo testuale "semplice" cioè codificato con l'uso del codice ASCII. Il nome "curriculum\_mario", più che avere valore per l'operazione di decodifica, ha valore di "descrittore del contenuto" ed è liberamente scelto dall'autore dell'oggetto come "promemoria":

in questo modo, chi si trovasse a leggere nell'elenco dei file:

*curriculum\_mario.txt*

*curriculum\_cesare.txt*

*curriculum\_rosa.txt*

dedurrebbe senza difficoltà il tipo e il contenuto dei file.

Nella seguente tabella si indicano le estensioni più utilizzate insieme al tipo di informazione che generalmente caratterizzano:

Tipologia	Estensioni
Grafica bitmap o raster	bmp, jpg, gif, tiff, png, ico
Grafica vettoriale	eps, ai, wmf
File eseguibili, codice oggetto, librerie dinamiche	exe, com, lib, jar, war, ear, class, dll
Documenti	doc, docx, html, rtf, odt, txt
Audio e musica	wav, wma, au, mp2, mp3, mid
Archiviazione e compressione	zip, 7z, jar, arc, gzip, tar, tar.gz, rar
Linguaggi di programmazione	c, cpp, for, java, lisp, cob, asm, bas, php, asp, jsp, pl, py, rb
Linguaggi di descrizione e stilizzazione pagina	pdf, postscript, xsl-fo, css, xslt/xsl



## Laboratorio: Conversione con OpenOffice Calc

### Conversione dei numeri da un sistema numerico ad un altro

Ora proviamo a costruire un foglio di calcolo che mi permetta di convertire i numeri nei vari sistemi numerici. Come ipotesi di partenza per tutti gli esercizi che realizzeremo, consideriamo sempre un numero con massimo 6 cifre, ma tale foglio di calcolo potrà essere ampliato a vostro piacimento utilizzando un numero diverso di cifre.

#### Conversione Binario → Decimale

Innanzitutto apriamo il foglio di calcolo e scriviamo le seguenti informazioni:

Con lo schema impostato come figura riusciremo a convertire il numero binario "011101" in un numero decimale.

	A	B	C	D	E	F	G	H
1	CONVERSIONE DI UN NUMERO							
2	da:							a:
3	BINARIO						DECIMALE	
4	0	1	1	1	0	1		
5	5	4	3	2	1	0	POSIZIONE	
6	2	2	2	2	2	2	BASE	
7							POTENZA	
8							PRODOTTO	



	A	B	C	D	E	F	G	H
1	CONVERSIONE DI UN NUMERO							
2	da:						a:	
3	BINARIO						DECIMALE	
4	0	1	1	1	0	1		29
5	5	4	3	2	1	0	0	POSIZIONE
6	2	2	2	2	2	2	2	BASE
7	32	16	8	4	2	1		POTENZA
8	0	16	8	4	0	1		PRODOTTO

Per prima cosa inserire tutte le informazioni così come scritte in figura.

Nella cella A7 scrivere la formula =A6^A5 e trascinare fino alla cella F7

Nella cella A8 scrivere la formula =A7\*A4 e trascinare fino alla cella F8

Nella cella H4 scrivere la formula =A8+B8+C8+D8+E8+F8 e premere INVIO

Ora sappiamo che "011101" corrisponde a "29" nel sistema decimale!

### Conversione Ottale → Decimale

Proviamo ora ad impostare un foglio di calcolo per effettuare la conversione da un numero ottale in decimale:

Per esempio, vogliamo trovare il corrispondente numero decimale del numero "256" in base 8.

Partiamo sempre costruendo una tabella come la figura seguente:

	A	B	C	D	E	F	G	H
1	CONVERSIONE DI UN NUMERO							
2	da:						a:	
3	OTTALE						DECIMALE	
4				2	5	6		
5	5	4	3	2	1	0	0	POSIZIONE
6	8	8	8	8	8	8	8	BASE
7								POTENZA
8								PRODOTTO

	A	B	C	D	E	F	G	H
1	CONVERSIONE DI UN NUMERO							
2	da:						a:	
3	OTTALE						DECIMALE	
4				2	5	6		174
5	5	4	3	2	1	0	0	POSIZIONE
6	8	8	8	8	8	8	8	BASE
7	32	16	8	4	2	1		POTENZA
8	0	0	0	128	40	6		PRODOTTO

Successivamente inserite le seguenti informazioni:

Nella cella A7 scrivere la formula =A6^A5 e trascinare fino alla cella F7

Nella cella A8 scrivere la formula =A7\*A4 e trascinare fino alla cella F8

Nella cella H4 scrivere la formula =A8+B8+C8+D8+E8+F8 e premere INVIO.

Ora sappiamo che il numero "256" in base 8 corrisponde al numero 174 nel sistema decimale.

### Conversione Esadecimale → Decimale

Per convertire un numero in sistema esadecimale in decimale, le cose si complicano un poco perchè le cifre del sistema esadecimale non sono tutte numeriche, ma dopo la cifra 9 dobbiamo considerare A, B, C, D, E, F che rispettivamente corrispondono a 10, 11, 12, 13, 14, 15. Procediamo quindi nel seguente modo:

Per prima cosa predisponiamo nel nostro foglio Calc nelle due colonne J e K l'associazione simbolo esadecimale ↔ valore



Successivamente impostiamo la seguente tabella:

	J	K		A	B	C	D	E	F	G	H
	0	0	1	1							
	1	1	2	2							
	2	2	3	3							
	3	3									
	4	4									
	5	5									
	6	6									
	7	7									
	8	8									
	9	9									
	A	10									
	B	11									
	C	12									
	D	13									
	E	14									
	F	15									

	A	B	C	D	E	F	G	H	
1	CONVERSIONE DI UN NUMERO								
2	da:							a:	
3	ESADECIMALE							DECIMALE	
4				1	4	A			
5							NUMERO CORRISPONDENTE		
6	5	4	3	2	1	0	POSIZIONE		
7	16	16	16	16	16	16	BASE		
8							POTENZA		
9							PRODOTTO		

Nelle celle A5:F5 dobbiamo impostare una Funzione che mi permetta di trovare la cifra numerica corrispondente alla cifra esadecimale inserita. La funzione in questione è **=CERCA.VERT()**. Come funziona? Questa funzione, in generale, permette di visualizzare un dato che si trova sullo stesso rigo di un valore dato in input. La sintassi della funzione è: **=CERCA.VERT(valore da cercare; tabella dei valori; colonna dove si trova il valore da visualizzare)**.

Quindi, il valore da cercare lo abbiamo inserito nelle caselle gialle; la tabella dei valori si trova nelle celle J1:K16 e l'informazione che ci serve si nella colonna 2 della stessa tabella J1:K16.

Nella cella A5, quindi, proviamo ad inserire la funzione: **=CERCA.VERT(A4;\$J\$1:\$K\$16;2)** e trascinatela fino alla cella F5.  
 Nella cella A8 scrivere la formula **=A7^A6** e trascinare fino alla cella F8  
 Nella cella A9 scrivere la formula **=A8\*A5** e trascinare fino alla cella F9  
 Nella cella H4 scrivere la formula **= A8+B8+C8+D8+E8+F8** e premere INVIO.

	A	B	C	D	E	F	G	H	
1	CONVERSIONE DI UN NUMERO								
2	da:							a:	
3	ESADECIMALE							DECIMALE	
4				1	4	A		330	
5	0	0	0	1	4	10	NUMERO CORRISPONDENTE		
6	5	4	3	2	1	0	POSIZIONE		
7	16	16	16	16	16	16	BASE		
8	1048576	65536	4096	256	16	1	POTENZA		
9	0	0	0	256	64	10	PRODOTTO		

Ora sappiamo che il numero in base esadecimale "14A" corrisponde al numero decimale "330".

**Conversione Decimale → Binario**

Ora proviamo a fare il contrario. Conoscendo il valore in decimale vogliamo calcolare il corrispondente numero in binario. Visto che abbiamo già utilizzato il numero 29 nel primo esempio, proviamo a prendere in considerazione questo numero per effettuare una controprova.

Impostare il foglio secondo la seguente figura:

Nella cella A7 scriviamo =A4  
 Nella cella C7 scriviamo =A7/B7  
 Nella cella D7 scriviamo =INT(C7)  
 Nella cella E7 scriviamo  
 =RESTO(A7;B7)

	A	B	C	D	E
1	CONVERSIONE DI UN NUMERO				
2					
3	DECIMALE		BINARIO		
4	29				
5					
6	DIVIDENDO	DIVISORE	RISULTATO	PARTE INTERA	RESTO
7			2		
8					
9					

se avete fatto tutto bene, otterrete il risultato come in figura:

	A	B	C	D	E
1	CONVERSIONE DI UN NUMERO				
2					
3	DECIMALE		BINARIO		
4	29				
5					
6	DIVIDENDO	DIVISORE	RISULTATO	PARTE INTERA	RESTO
7	29	2	14,5	14	1
8					
9					

Ora nella cella A8 scriviamo =D8  
 Nella cella C8 scriviamo =A8/B8  
 Nella cella D8 scriviamo =INT(C8)  
 Nella cella E8 scriviamo =RESTO(A8;B8)

Ripetiamo la stessa operazione fino a quando la cella che contiene la "Parte Intera" raggiungerà il valore "0", come la seguente figura:

Ora bisogna fare in modo che resti delle varie divisioni vengano letti a partire dall'ultimo.

	A	B	C	D	E
1	CONVERSIONE DI UN NUMERO				
2					
3	DECIMALE		BINARIO		
4	29				
5					
6	DIVIDENDO	DIVISORE	RISULTATO	PARTE INTERA	RESTO
7	29	2	14,5	14	1
8	14	2	7	7	0
9	7	2	3,5	3	1
10	3	2	1,5	1	1
11	1	2	0,5	0	1

Possiamo usare per esempio la funzione =CONCATENA() che permette di unire in un'unica cella informazioni che si trovano in più celle.

Proviamo a scrivere nella cella C4 la seguente funzione =CONCATENA(E11;E10;E9;E8;E7).

Otterremo la seguente figura:

	A	B	C	D	E
1	<b>CONVERSIONE DI UN NUMERO</b>				
2					
3	<b>DECIMALE</b>		<b>BINARIO</b>		
4	29		11101		
5					
6	<b>DIVIDENDO</b>	<b>DIVISORE</b>	<b>RISULTATO</b>	<b>PARTE INTERA</b>	<b>RESTO</b>
7	29	2	14,5	14	1
8	14	2	7	7	0
9	7	2	3,5	3	1
10	3	2	1,5	1	1
11	1	2	0,5	0	1

Abbiamo, in questo modo, verificato che effettivamente il numero decimale "29" corrisponde al numero binario "11101".

### Conversione Decimale → Ottale

Per quanto riguarda l'impostazione del foglio per eseguire questa conversione procediamo come nell'esercizio precedente, solo che questa volta nella colonna DIVISORE dobbiamo digitare "8".

Otterrete il seguente risultato:

	A	B	C	D	E
1	<b>CONVERSIONE DI UN NUMERO</b>				
2					
3	<b>DECIMALE</b>		<b>BINARIO</b>		
4	174		00256		
5					
6	<b>DIVIDENDO</b>	<b>DIVISORE</b>	<b>RISULTATO</b>	<b>PARTE INTERA</b>	<b>RESTO</b>
7	174	8	21,75	21	6
8	21	8	2,63	2	5
9	2	8	0,25	0	2
10	0	8	0	0	0
11	0	8	0	0	0

### Conversione Decimale → Esadecimale

Anche in questo caso procedete sempre allo stesso modo, modificando semplicemente la colonna relativa a divisore, inserendo questa volta il valore "16".

Otterrete il seguente risultato:

	A	B	C	D	E
1	<b>CONVERSIONE DI UN NUMERO</b>				
2					
3	<b>DECIMALE</b>		<b>BINARIO</b>		
4	330				
5					
6	<b>DIVIDENDO</b>	<b>DIVISORE</b>	<b>RISULTATO</b>	<b>PARTE INTERA</b>	<b>RESTO</b>
7	330	16	20,63	20	10
8	20	16	1,25	1	4
9	1	16	0,06	0	1
10	0	16	0	0	0
11	0	16	0	0	0

Per poter assegnare la lettera per i numeri maggiori di 9, bisogna utilizzare sempre il comando =CERCA.VERT() impostando a parte una tabella simile a quella utilizzata nell'esercizio di conversione da esadecimale a decimale.

Nella cella C4 bisognerà utilizzare la funzione =CONCATENA(E7;G1:H16;2). Otterremo, in questo modo la seguente figura:

	A	B	C	D	E	F	G	H
1	CONVERSIONE DI UN NUMERO							
2							0	0
3	DECIMALE		BINARIO				2	2
4	330						3	3
5							4	4
6	DIVIDENDO	DIVISORE	RISULTATO	PARTE INTERA	RESTO		5	5
7	330	16	20,63	20	10	A	6	6
8	20	16	1,25	1	4	4	7	7
9	1	16	0,06	0	1	1	8	8
10	0	16	0	0	0		9	9
11	0	16	0	0	0		10	A
12							11	B
13							12	C
14							13	D
15							14	E
16							15	F

anteprima 50 pagine

### 3. I connettivi logici dell'Algebra di Boole



Autore: Angelo Oliva

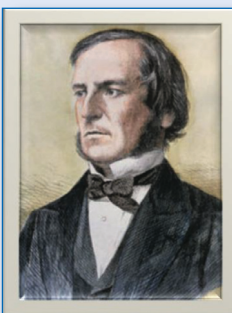
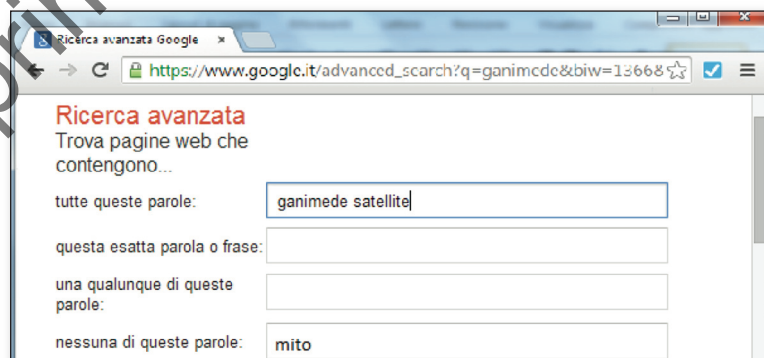
Competenze	Abilità	Conoscenze
<i>Interpretare situazioni complesse, composte da più condizioni, comprenderne il significato.</i>	<i>Risolvere espressioni logiche a due e tre variabili, formulare proposizioni composte per selezionare particolari sott'insiemi di elementi su gruppi complessi</i>	<i>Elementi delle algebre di Boole e della logica delle proposizioni</i>

#### Introduzione

Nell'ambito dell'informatica in generale e della programmazione in particolare non si può prescindere dalla conoscenza dei principi della logica degli enunciati. Ad esempio, quando un'azione si può svolgere solo se si verificano due condizioni (se ho l'automobile e le catene a bordo potrò percorrere la strada innevata).

L'applicazione dei principi della logica degli enunciati non è prerogativa esclusiva della programmazione, ne è prova l'inserimento di tale argomento nel programma ufficiale per il conseguimento della patente europea per il computer, meglio nota con l'acronimo inglese ECDL e precisamente nel primo dei sette moduli del syllabus Ver. 5.0, Concetti di Base dell'ICT, paragrafo 1.0.3.2 "Distinguere il ruolo dei connettivi logici (AND, OR, NOT) nell'informatica."

In passato l'uso dei connettivi logici era necessario per svolgere delle ricerche avanzate su internet con i motori più conosciuti, per esempio per cercare informazioni su Ganimede inteso come pianeta e non come figura mitologica bisognava digitare nella barra d'inserimento della chiave di ricerca "Ganimede AND satellite NOT mito". Oggi, per semplificare le ricerche avanzate, i motori di ricerca prevedono la possibilità di perfezionare le ricerche attraverso un menù che facilita la costruzione della chiave di ricerca.



George Boole (1815 - 1864)

Nel 1854 pubblicò la sua opera più importante, "An investigation into the Laws of Thought, on Which are founded the Mathematical Theories of Logic and Probabilities" indirizzata alle leggi del pensiero, con la quale propose una nuova impostazione della logica: scopo dell'opera fu di studiare le leggi delle operazioni mentali alla base del ragionamento esprimendole nel linguaggio simbolico del calcolo e di istituire, di conseguenza, una disciplina scientifica della logica sorretta da un metodo; dopo aver rilevato le analogie fra oggetti dell'algebra e oggetti della logica, ricondusse le composizioni degli enunciati a semplici operazioni algebriche. Con questo lavoro fondò la teoria di quelle che ora sono dette **algebre di Boole** (o semplicemente **algebra booleana**). Pur mantenendo distinte le operazioni mentali da quelle algebriche e le leggi logiche dai settori delle scienze naturali, il compito di Boole fu quello di travestire la logica con un abito matematico algebrico. I così detti "valori Booleani", sono utilizzati molto frequentemente nell'informatica e ne sono essenziali, dai linguaggi di basso livelli (**Assembly**), a quelli di alto livello e alle tecnologie web (**php**).



## Le Proposizioni e i predicati

La logica delle proposizioni è un linguaggio formale regolato da una semplice struttura sintattica, basata su proposizioni elementari (atomi) che composte tramite connettivi logici restituiscono un valore di verità in base ai singoli valori di verità delle proposizioni connesse.

La Proposizione nell'ambito della logica delle proposizioni si distingue da quella del dizionario perché si definisce Proposizione (o Enunciato) una qualsiasi espressione autonoma e di senso compiuto alla quale si può attribuire un valore di verità Vero o Falso in modo oggettivo.

“Sta piovendo”, “quella macchina è rossa” sono proposizioni valide perché si può affermare in modo incontrovertibile se sono vere o false, invece le espressioni “non fa molto freddo” o “il colore più bello è il rosso” pur potendo essere vere o false non sono oggettive ma dipendono dal parere del soggetto che le valuta.

Il Predicato è una proposizione contenente una o più variabili il cui valore di verità in un determinato istante dipende dai valori assunti dalle sue variabili nello stesso istante.

Partendo dalla proposizione esempio “Taranto è una provincia della Puglia” si possono fare degli esempi di predicati:

“X è provincia di Puglia” con X appartenente all'insieme delle città d'Italia.

“X è provincia di Y” con X appartenente all'insieme delle città d'Italia e Y appartenente all'insieme delle Regioni.

Se il valore di X è Roma, il primo predicato assumerà valore falso, se Y è Lazio il secondo predicato, alla presenza dello stesso valore di X, assumerà valore Vero.

Esercitazione:

4. Quali delle seguenti frasi di senso compiuto è una proposizione?

- Lorenzo Giovanotti è un cantante
- Londra è la capitale del Brasile
- Non c'è festa più bella del Natale
- Il colore più bello è il viola
- Oggi in classe ci sono più assenti di ieri
- Quelle scarpe sono costose
- Tre è pari
- Febbraio ha 30 giorni

5. Trasformate la proposizione “Euro è la valuta dell'Italia”,

- in un predicato ad una variabile (R1)
- in un predicato a due variabili (R2)



## Composizione di proposizioni semplici.

Le Proposizioni si dicono **Composte** quando sono formate da proposizioni semplici collegate da connettivi logici. Per esempio “la temperatura è zero gradi **E** sta nevicando” è una proposizione composta tramite una congiunzione logica (la **E** della frase che come vedremo tra poco corrisponde formalmente al connettivo logico **AND**), il valore di verità della proposizione composta dipenderà dai valori di verità delle proposizioni semplici e dal connettivo logico che li unisce, cioè nel caso che sia vero che la temperatura sia di zero gradi ma non sia vero che sta nevicando, la proposizione composta tramite la congiunzione logica ha valore di verità falso.