

Flash card

IL PENSIERO COMPUTAZIONALE



Figura A
Programma il Futuro
1. Fai clic e scegli Iscrizione per studenti/CLICCA QUI PER ISCRIVERTI per registrarti e poi eseguire l'accesso.

SEGUIMI!

L'espressione **Pensiero computazionale** è stata coniata nel 2006 dalla scienziata informatica Jeannette Wing.

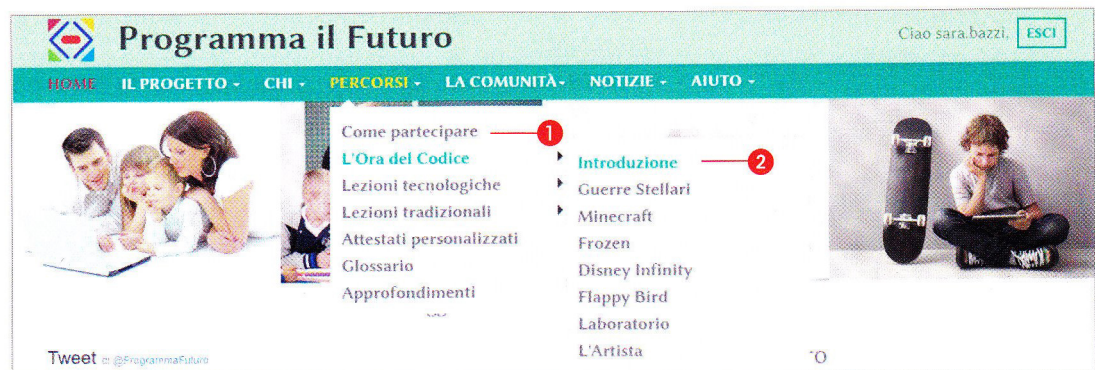


Figura B
Percorsi
1. Attiva il link Come partecipare e guarda i video di esempio disponibili.
2. Da L'Or del Codice scegli Introduzione per accedere ai diversi percorsi.

Ogni giorno, quando affronti una qualsiasi delle tue attività, in realtà, senza renderti conto, risolvi dei "problemi". Per esempio, il semplice fatto di recarti a scuola, presuppone che tu abbia definito il percorso (imboccando una strada piuttosto che un'altra, svoltando a destra perché a sinistra ci sono dei lavori in corso, attraversando una strada ecc.) che ritieni più idoneo per raggiungere il tuo obiettivo: arrivare a scuola. Il **pensiero computazionale** si focalizza proprio su questo: formulare una procedura che, eseguita in un determinato contesto, consenta di raggiungere l'obiettivo prefissato.

Il sito www.programmailfuturo.it fornisce la seguente definizione: "Con il pensiero computazionale si definiscono **procedure** che vengono poi attuate da un **esecutore**, che opera nell'ambito di un **contesto prefissato**, per raggiungere degli **obiettivi assegnati**". Il video, disponibile nel medesimo sito all'indirizzo www.programmailfuturo.it/progetto/cose-il-pensiero-computazionale, dà un'idea immediata e concreta del concetto.

Allenarsi al pensiero computazionale addestra a risolvere, anche in team, problemi complessi, formulando soluzioni che indicano in modo chiaro e preciso le diverse fasi per raggiungere l'obiettivo (come quando costruisci un algoritmo). Come si

Programma il Futuro Ciao sara.bazzi ESCI

HOME IL PROGETTO CHI PERCORSI LA COMUNITA NOTIZIE AIUTO

La puoi svolgere con una lezione *tradizionale*, denominata *Pensiero Computazionale*, oppure con una qualunque di queste 8 lezioni *tecnologiche* (vedi più oltre le indicazioni per scegliere il corso appropriato all'età degli studenti):

1. *Guerre Stellari*, con l'ambientazione della saga cinematografica di Guerre Stellari - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte; questa Ora del Codice è disponibile anche nella [versione in JavaScript](#), per studenti delle superiori, in cui si può alternare programmazione a blocchi e programmazione testuale;
2. *Minecraft*, con l'ambientazione del video-gioco Minecraft - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
3. *Il Labirinto*, con i personaggi dei giochi "Angry Birds" e "Zombie vs Plants", e con Scrat del film "L'era glaciale" - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte; questa Ora del Codice è disponibile anche nella [versione del 2013](#), con i personaggi dei giochi "Angry Birds" e "Zombie vs Plants" - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
4. *Frozen*, con Anna ed Elsa del film "Frozen" - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
5. *Disney Infinity*, per creare una storia o inventare un gioco con i personaggi di Disney Infinity - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
6. *Flappy Bird*, per costruire la tua versione di questo gioco - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
7. *Laboratorio*, per creare una storia o inventare un gioco - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte;
8. *L'Artista*, per costruire fantastici disegni e figure - è disponibile una [pagina con video tutorial](#) che descrive le attività svolte.

Per chi non ha mai interagito con un computer è disponibile questa [esercitazione introduttiva](#) per imparare come cliccare, trascinare e rilasciare.

Come guida alla selezione della migliore versione dell'Ora del Codice suggeriamo quanto segue, in funzione del livello della classe e assumendo che gli studenti non abbiano precedenti esperienze di programmazione:

- *Prima e seconda elementare*: Corso 1 (svolgere la [lezione 4](#) e la [lezione 5](#), se c'è tempo la [lezione 13](#));
- *Terza elementare*: Corso 2 (svolgere la [lezione 3](#) e la [lezione 8](#), se c'è tempo la [lezione 13](#));
- *Quarta e quinta elementare*: *Minecraft* oppure *Frozen* oppure Corso 2 (svolgere la [lezione 3](#) e la [lezione 8](#) e la [lezione 11](#));
- *Medie*: *Guerre Stellari* oppure *Disney Infinity* oppure *Flappy* oppure Corso 3 (svolgere la [lezione 3](#) e la [lezione 7](#) e la [lezione 8](#));
- *Superiori*: [Il Labirinto](#) oppure *Laboratorio* oppure *L'Artista* oppure Corso 4 (svolgere la [lezione 2](#), la [lezione 7](#) e la [lezione 9](#)) oppure *Guerre Stellari* nella [versione in JavaScript](#).

Figura C
Pagina Introduzione
Elenca i percorsi disponibili (ti consigliamo di scegliere Il Labirinto).

SEGUIMI!

Ricorda di eseguire l'accesso anche in [Code.org](#) se vuoi **ottenere**, al termine del percorso, il relativo **attestato**.

Figura D
Accesso a Code.org
Elenca i percorsi disponibili (ti consigliamo di scegliere Il Labirinto).

Figura E
Sezione L'Ora del Codice in Code.org
Scegli Labirinto Classico.

può approfondire attivando il link precedente, si tratta di un processo mentale che combina **metodi caratteristici e strumenti intellettuali** di valore generale.

Dal sito [www.programmailfuturo.it](#) puoi accedere a diversi percorsi, resi disponibili sul sito di fruizione [Code.org](#), che allenano al pensiero computazionale. Dopo esserti registrato (*Figura A*) e aver eseguito il login, attiva il link [Percorsi](#) (*Figura B*). Dalla pagina [Introduzione](#) (*Figura C*) ti consigliamo di scegliere il percorso [Il Labirinto](#) e poi [Accedi alla lezione](#); entri in questo modo nel sito di fruizione [Code.org](#), nel quale ti raccomandiamo di inserire i dati (*Figura D*) con i quali ti sei registrato in [Programma il Futuro](#) (in modo che tutta la tua attività venga registrata e tu possa, al termine, ottenere l'attestato), e poi conferma la scelta facendo clic su [Labirinto Classico](#) (*Figura E*) per iniziare il percorso.

Le attività proposte non sono banali, come potrebbe apparire a un primo approccio, perché abitano la mente a inventare e poi definire precise procedure per risolvere le diverse situazioni che si presentano, come accade nella quotidianità, in momenti di vita personale o professionale.

Flash card

IL PENSIERO COMPUTAZIONALE

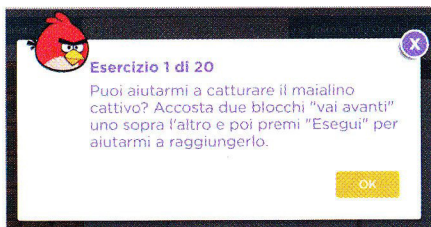


Figura F
Box richiesta Esercizio 1

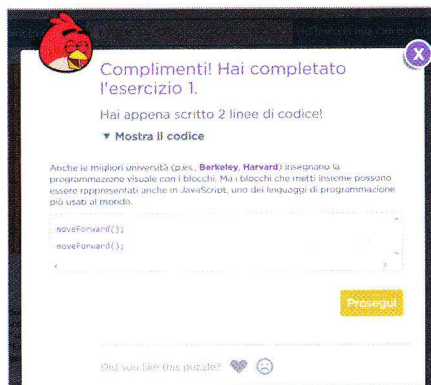


Figura H
Feedback Esercizio 1



Figura I
Esempio di attestato L'Or del Codice

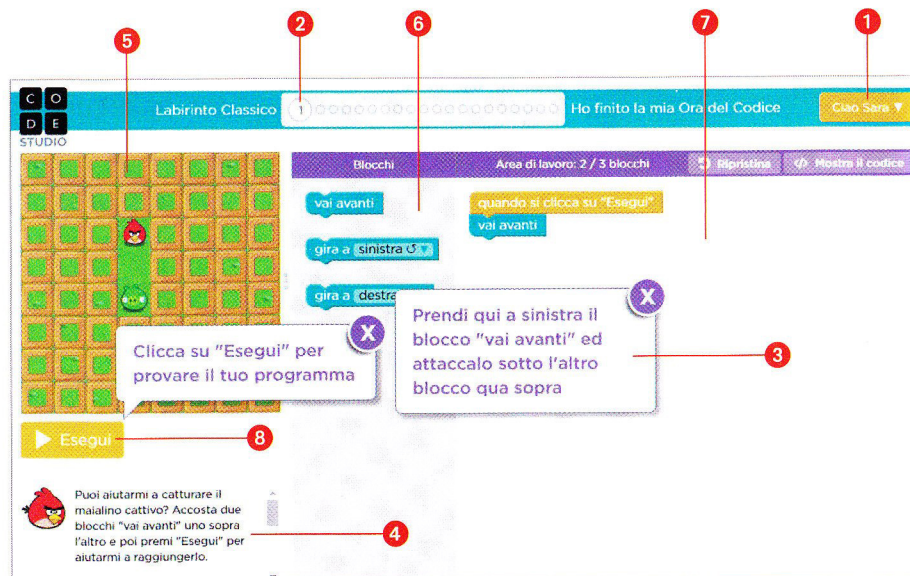


Figura G Esercizio 1

- 1 Per eseguire l'accesso e il logout
- 2 Esercizio in atto (1 di 20)
- 3 Box istruzioni
- 4 "Problema" da risolvere
- 5 Area Studio, con la grafica della situazione proposta
- 6 Area Blocchi, con i blocchi disponibili

- 7 Area di lavoro in cui aggiungere i blocchi per risolvere il problema e raggiungere l'obiettivo.
- 8 Esegue le istruzioni, costituite dalla combinazione dei blocchi collocati nell'Area di lavoro, in modo da verificare la correttezza della soluzione ideata.

SEGUIMI!

Puoi **completare il percorso in momenti diversi**. Per tornare a un esercizio interrotto e proseguire, puoi **accedere direttamente al sito di fruizione <https://code.org/>**, fai clic su **Accedi**, inserisci le tue credenziali di registrazione e poi conferma con **Accedi**. Nella pagina caricata (Figura J) puoi scegliere se proseguire con il successivo esercizio, oppure se visualizzare il corso (Figura K) per avere un'idea del punto a cui ti trovi.

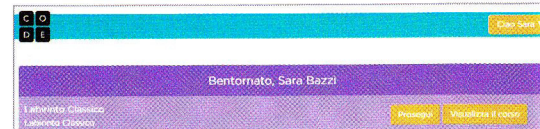


Figura J Accesso all'area personale di Code.org

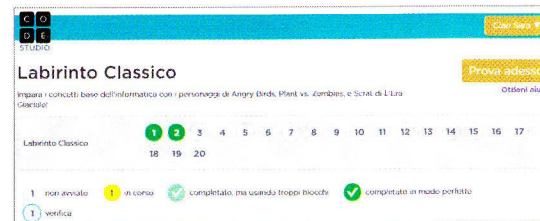


Figura K Visualizza il corso

Dopo aver visionato il video introduttivo, compare la prima richiesta (Figura F) e, facendo clic su **OK**, l'ambiente in cui costruire la soluzione (Figura G) e verificarla. Ottieni subito un feedback (Figura H) e, facendo clic su **Mostra il codice**, puoi leggere come sono rappresentati nel linguaggio di programmazione JavaScript i blocchi che hai utilizzato. Fai clic su **Prosegui** per avviare il secondo esercizio. Al termine dei 20 esercizi ottieni l'attestato, **Certificate of Completion The Hour of Code**, completo del tuo Nome e Cognome (Figura I).

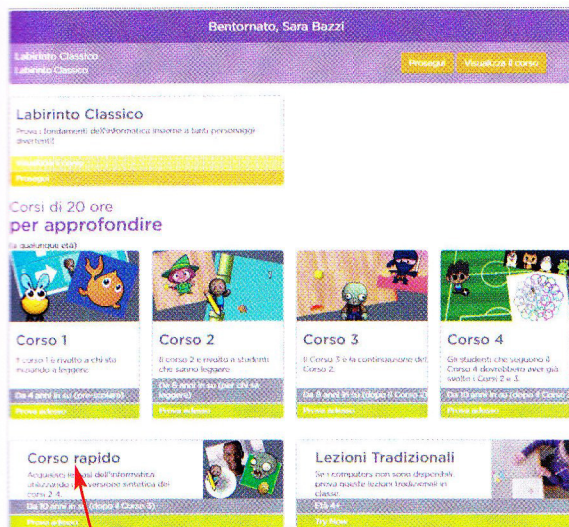


Figura L
Alcuni dei corsi disponibili
in Code.org

Figura M
Esempio
di attestato
Corso rapido

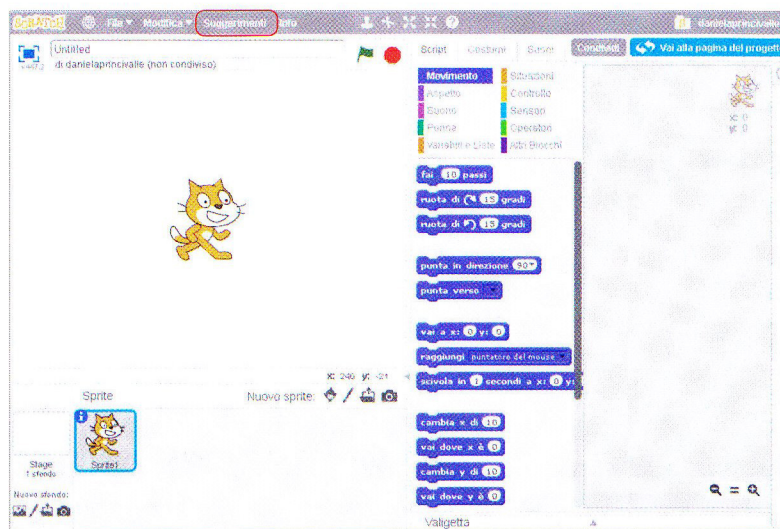


Figura N
Ambiente Scratch

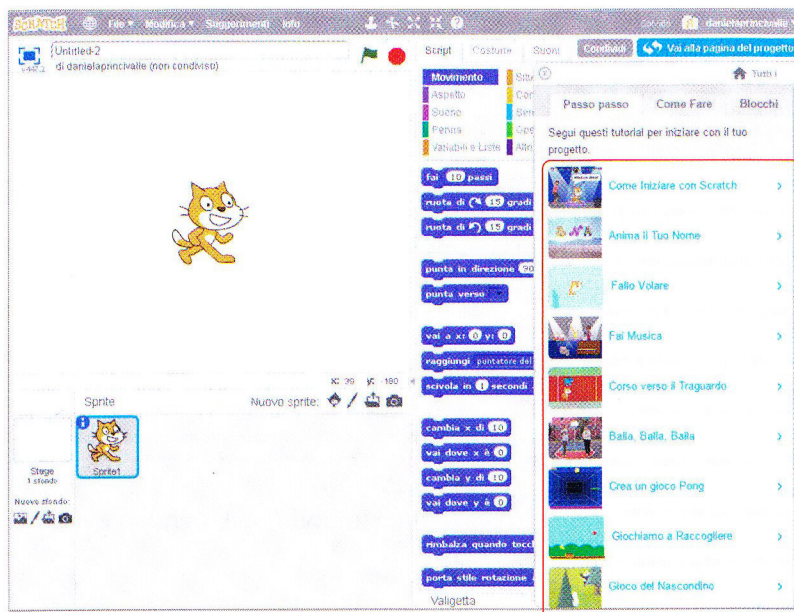


Figura O
Elenco di tutorial

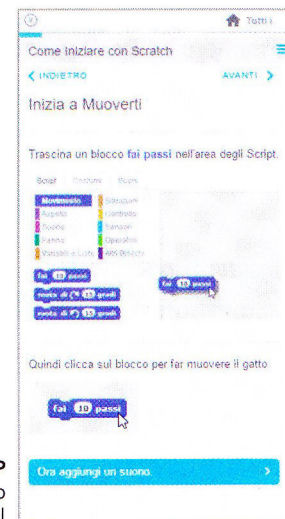
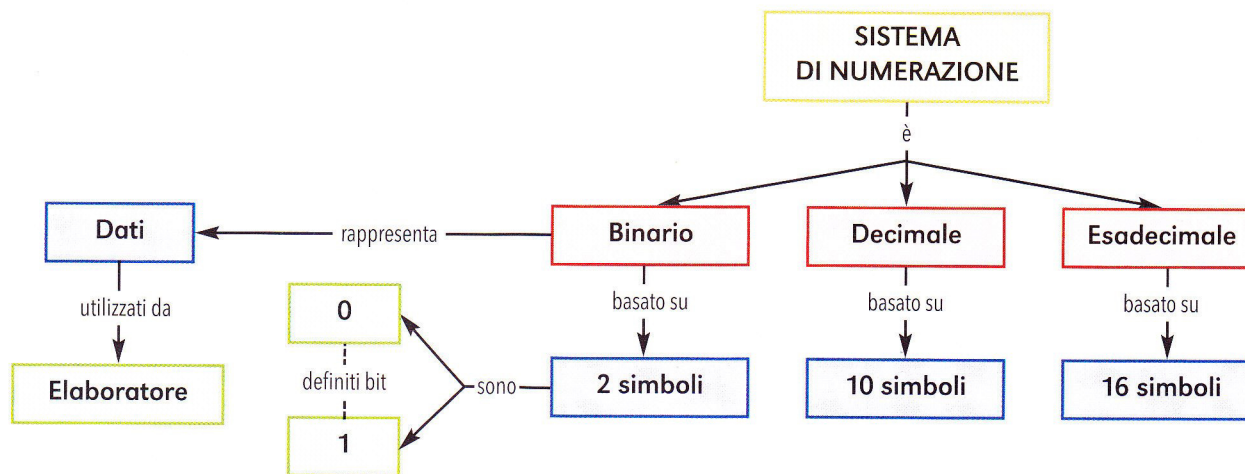


Figura P
Esempio
di tutorial

Concluso il percorso **L'Ora del Codice**, dal sito **Code.org** ti suggeriamo di svolgere il **Corso rapido** (Figura L), il cui tempo di esecuzione totale è stimato in 20 ore, alla fine del quale ottieni un ulteriore attestato (Figura M).

Un **linguaggio di programmazione visuale** che offre un ambiente che ricorda quello proposto nei percorsi **Code.org** è **Scratch** (Figura N). Facendo clic su **Suggerimenti**, si ottiene un elenco di tutorial (Figura O) che, mediante animazioni, guidano passo passo all'uso (Figura P).

I sistemi di numerazione decimale, binario, esadecimale



Esistono diversi sistemi di numerazione, e qui di seguito prenderemo in considerazione quelli **posizionali decimale, binario ed esadecimale**, nei quali i simboli di cui si compongono cambiano di valore in base alla loro posizione.

Il sistema di numerazione decimale si basa su dieci simboli, dallo 0 al 9.

ESEMPIO 5

Se scriviamo 584, oppure 845, utilizziamo le medesime cifre che però, modificando la loro posizione, assumono un valore diverso: spostando una cifra verso sinistra si moltiplica il suo valore per dieci, mentre spostandola di una posizione verso destra si divide il suo valore per dieci.

Ogni numero intero N può essere scritto nella seguente forma:

$N = a_n a_{n-1} a_{n-2} a_{n-3} \dots a_2 a_1$ dove a_k rappresenta una qualsiasi delle cifre da 0 a 9 ed equivale all'espressione

$N = a_n * 10^{n-1} + a_{n-1} * 10^{n-2} + \dots + a_2 * 10^1 + a_1 * 10^0$ polinomio in cui ogni cifra viene rappresentata moltiplicandola per una **potenza di 10**. I simboli a_k sono le **cifre del numero N e 10 la sua base**.

In considerazione di ciò, i numeri 584 e 845 possono essere rappresentati nella forma mostrata a lato, in cui ogni cifra, in base alla posizione occupata, è stata moltiplicata per la potenza di dieci.

$$584 = 5 * 10^2 + 8 * 10^1 + 4 * 10^0$$

$$845 = 8 * 10^2 + 4 * 10^1 + 5 * 10^0$$

Il sistema di numerazione binario si basa su due simboli 0 e 1, ognuno dei quali prende il nome di bit (binary digit, cifra binaria).

ESEMPIO 6

Quando si desidera accendere la luce in una stanza si preme l'interruttore che fa accendere la lampadina. In questo caso si hanno due soli possibili stati: con il circuito elettrico chiuso la lampadina è accesa, con il circuito elettrico aperto la lampadina è spenta.

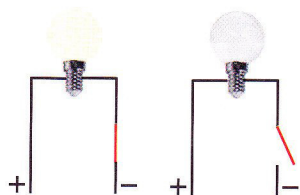


Figura 5
Esempio di circuito chiuso e aperto

L'esempio appena illustrato ci è utile in quanto è paragonabile al modo in cui lavorano i computer, i quali utilizzano circuiti elettronici che prevedono **solo due possibili stati** (circuito **aperto**/circuito **chiuso** (Figura 5), oppure **magnetizzato/non magnetizzato**): entrambe queste situazioni, per essere rappresentate in modo matematico, ricorrono al sistema binario.

AL VOLO!

Il numero decimale 10 corrisponde al numero binario:

- 1010
- 0101

Anche in questo sistema di numerazione, come in quello decimale, spostando una cifra verso sinistra si moltiplica il suo valore per due, mentre spostandola di una posizione verso destra si divide il suo valore per due: per tale motivo, anche ai numeri binari si associa la potenza di 2 in corrispondenza della posizione occupata.

Convertire un numero binario in decimale

Per convertire un **numero binario in decimale**, partendo da destra, si moltiplica ogni cifra per la potenza di 2 corrispondente alla posizione occupata. Il pedice 2 indica un numero espresso in sistema binario e il pedice 10 un numero espresso in sistema decimale.

ESEMPIO 7

Convertiamo il numero decimale binario 1101001 in decimale.

$$\begin{aligned} 1101001_2 &= 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 1 \cdot 64 + 1 \cdot 32 + 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 \\ &= 64 + 32 + 0 + 8 + 0 + 0 + 1 \\ &= 105_{10} \end{aligned}$$

SEGUIMI!

Vedi l'applicazione di questo calcolo con Excel a pag. 209

SEGUIMI!

Il **punto radice** separa, come nel consueto uso della virgola, la parte intera del numero da quella non intera.

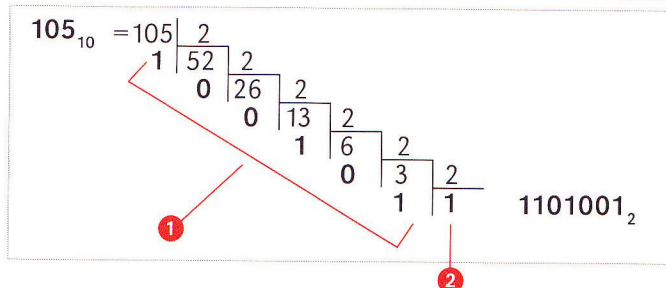
Convertire un numero decimale in binario

Per convertire un **numero decimale in binario**, si divide il numero per 2 e il resto rappresenta la cifra meno significativa in binario; si divide poi per 2 il quoziente ottenuto e il resto rappresenta la seconda cifra del **punto radice** e così si procede fino all'ultimo quoziente 1. Nella lettura del numero binario si parte dall'ultimo quoziente 1 e poi si leggono i resti ottenuti.

ESEMPIO 8

Convertiamo il numero decimale 105 in binario.

- 1 Resti ottenuti
- 2 Ultimo quoziente



Confrontando la **rappresentazione del valore** considerato nei due sistemi, si può notare che, **con il sistema binario, più semplice rispetto al decimale**, si deve utilizzare un numero maggiore di cifre. Tuttavia, semplicità e numero elevato di calcoli sono caratteristiche che ben si associano alla logica di lavoro di un elaboratore, in grado di compiere un numero elevatissimo di operazioni in pochissimo tempo.

Per rappresentare in modo più sintetico gruppi di bit, gli elaboratori fanno ricorso anche al sistema di numerazione esadecimale.

Il sistema di numerazione esadecimale si basa su 16 simboli, dallo 0 alla F.

Nella seguente tabella riportiamo i simboli del sistema esadecimale al loro valore in decimale.

Esadecimale	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimale	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Convertire un numero decimale in esadecimale e viceversa

Per convertire un **numero da esadecimale a decimale**, e viceversa, si procede analogamente a quanto visto per il sistema binario.

ESEMPIO 9

Convertiamo il numero esadecimale D3F in decimale.

$$\begin{aligned} D3F_{16} &= D \cdot 16^2 + 3 \cdot 16^1 + F \cdot 16^0 \\ &= 13 \cdot 256 + 3 \cdot 16 + 15 \cdot 1 \\ &= 3328 + 48 + 15 \\ &= 3391_{10} \end{aligned}$$

AL VOLO!

Il numero esadecimale A1 corrisponde al numero decimale:

- 161
- 162

ESEMPIO 10

Convertiamo il numero decimale 58506 in esadecimale.

$$\begin{array}{r} 58506_{10} = 58506 \begin{array}{l} | \quad 16 \\ \hline 10 \quad 3656 \\ \quad | \quad 16 \\ \quad \quad 8 \quad 228 \\ \quad \quad | \quad 16 \\ \quad \quad \quad 4 \quad 14 \\ \quad \quad \quad \quad | \quad 16 \\ \quad \quad \quad \quad \quad 14 \end{array} \quad E48A_{16} \\ \downarrow \quad \quad \quad \downarrow \\ A \quad \quad \quad E \end{array}$$

AL VOLO!

Il numero decimale 16 corrisponde al numero esadecimale:

- 10
- 0101

Concetto di bit e di byte

Per poter rappresentare i caratteri in un sistema elettronico di elaborazione, le **due cifre** di cui si compone il sistema binario sarebbero tuttavia insufficienti: sono state quindi raggruppate in **byte**.

Il bit corrisponde a una delle due cifre binarie, **0** oppure **1**.

Il byte è un **gruppo di 8 bit**, unità minima di memorizzazione e lettura in un elaboratore, e serve per rappresentare una lettera, una cifra o un simbolo. Nella codifica delle informazioni si utilizzano il byte e i suoi multipli.

Con 1 byte si possono ottenere 256 (2^8) combinazioni di 0 e 1, ognuna delle quali, mediante una tabella di codifica, viene associata a una lettera dell'alfabeto, a un numero (da 0 a 9), a un carattere speciale e ad altri segni o simboli.

La **tabella standard di codifica** che permette tale collegamento è il **codice ASCII** (*American Standard Code for Information Interchange*), usato per la rappresentazione dei caratteri di testo attraverso i byte.

In realtà, il codice ASCII copre solo i primi 128 caratteri, numeri e simboli; i successivi, fino al 256, costituiscono la **tabella ASCII estesa**, che presenta varie versioni a carattere nazionale.

SEGUIMI!

Il termine **digitale** (dall'inglese *digit*, *cifra*) sta a indicare tutto ciò **che può essere rappresentato da numeri**.

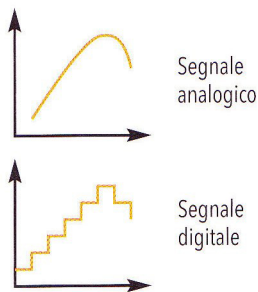


Figura 6
Rappresentazione grafica di segnale analogico e digitale

Rappresentazione digitale dei dati

Nella vita di ogni giorno alcune situazioni sono direttamente rappresentate da numeri; si pensi, per esempio, al numero di penne presenti nell'astuccio: considerato in modo **digitale**, si tratta di una grandezza "che varia fra un sistema finito di possibili stati in maniera discreta", dove "**stati**" indica nel nostro esempio il numero di penne e "**discreta**" significa che ogni stato (numero) è ben distinto dagli altri, vale a dire 4 penne, oppure 5 ecc., cioè valori ben definiti.

Altre situazioni, invece, come per esempio il segnale generato da un microfono, si presentano in modo analogico in quanto, in un intervallo, il segnale può assumere infiniti valori.

L'elaboratore è in grado di operare solo su dati digitali rappresentati da valori binari; un fenomeno analogico, dunque, per poter essere interpretato, deve essere digitalizzato (Figura 6): il suo originario stato analogico viene "tradotto" e rappresentato mediante un insieme numerabile di elementi.

SEGUIMI!

Le immagini digitali possono essere distinte in due principali categorie: **bitmap** e **vettoriali**.

- Immagini **bitmap**: sono **costituite da pixel**, il cui numero determina la qualità e il peso. Se ingrandite, risultano sfuocate ed è riconoscibile la quadrettatura (*blocking*) dei pixel (Figura 7).
- Immagini **vettoriali**: sono **costituite da linee e curve (vettori)**, a cui sono attribuiti i valori relativi alla posizione e al colore. Possono essere ingrandite senza perdita di qualità. Un formato di immagine vettoriale molto diffuso è **.svg**.

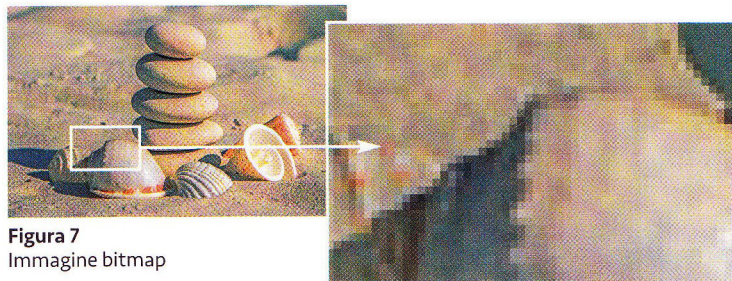


Figura 7
Immagine bitmap

SEGUIMI!

Con il termine **RGB** si intende un **modello cromatico** che si basa su tre colori: rosso (*Red*), verde (*Green*), blu (*Blue*). Mediante filtri o altre tecniche, un'immagine può essere scomposta in questi colori base che, miscelati tra loro, forniscono lo spettro dei colori visibili.

Una foto in formato digitale può essere divisa in un numero discreto di "punti", piccoli quadrati o rettangoli, detti **pixel** (dall'inglese *picture element*), ognuno dei quali è costituito da un colore tra i 16.777.216 possibili (se codificati nel modello di colori **RGB**).

Ogni pixel è formato da un determinato numero di byte, in relazione al colore che deve rappresentare: tali byte indicano l'intensità di rosso, quella di verde e quella di blu (RGB) che deve essere associata al relativo punto dell'immagine per poter essere rappresentato correttamente.

Ne consegue che un'immagine, in relazione al numero di pixel di cui si compone e al numero di byte di cui ogni pixel è costituito, può essere a risoluzione più o meno elevata. **Maggiore** sarà la **risoluzione** (ovvero il numero di pixel che forma l'immagine), **migliore** sarà la sua **rappresentazione** in termini di nitidezza e precisione e **più grandi** saranno le sue **dimensioni** in termini di byte (l'immagine sarà "**pesante**").

SEGUIMI!

Per ridurre le dimensioni del file spesso vengono applicate **tecniche di compressione** che si distinguono in:

- **lossy** (con perdita di dati), dove maggiore sarà il livello di compressione, minore sarà la qualità;
- **lossless** (senza perdita di dati), realizzata, per esempio nelle immagini, raggruppando sequenze di pixel uguali.

A lato sono mostrati alcuni formati associati al tipo di compressione applicata a file di immagini.

La grande diffusione di dispositivi di elaborazione che gestiscono informazioni in formato digitale ha fortemente favorito la traduzione in termini digitali di grandezze e fenomeni analogici.

Tecnica di compressione	Formato
lossy	jpeg
lossless	gif, png, bmp



Rifletto...



Osserva le pagine da 2 a 15, senza leggerne per intero il testo, poi rispondi alle seguenti domande.

A Qual è il titolo dell'UDA?

B Quali sono i titoli dei paragrafi?

.....
.....
.....
.....
.....
.....
.....

C Concentrati sul paragrafo "Il linguaggio naturale e i linguaggi di programmazione". Guardando la *Figura 4* si può vedere che, nel risolvere problemi, il linguaggio naturale, nella prima fase, viene espresso mediante:

.....

D Sempre osservando la *Figura 4*, la CPU quale linguaggio è in grado di comprendere?

.....

Costruisci il tuo glossario

TERMINE	SIGNIFICATO
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....



Organizzo...



Ora leggi la parte di testo indicata dal tuo insegnante ed evidenzia le informazioni più importanti, evitando parole superflue (avverbi, aggettivi, informazioni che già possiedi ecc.), poi:

- cerchi le parole chiave;
- collega con frecce le parole in relazione logica tra loro;
- scrivi qui sotto un titolo che indichi sinteticamente l'argomento centrale della parte di testo considerata, fai una mappa concettuale delle informazioni e un breve riassunto o un elenco.

TITOLO

.....

CREO LA MAPPA



RIASSUNTO/ELENCO

.....
.....
.....
.....
.....



Eseguo i test

CONVERTI E RISPONDI



Binario	Decimale	Esadecimale	Decimale
1000011		F9	
1011110		A2D	
Decimale	Binario	Decimale	Esadecimale
99		37	
124		735	

COMPLETAMENTO



classe - alto - due - macchina - sequenza - byte

- 1 L'algoritmo è una finita di istruzioni che consente di risolvere tutti i problemi di una determinata e produrre il risultato stabilito.
- 2 Il bit corrisponde a una delle cifre binarie. Un gruppo di 8 bit forma un
- 3 Il linguaggio di programmazione è definito anche linguaggio di livello.
- 4 Il linguaggio è l'unico direttamente comprensibile dall'elaboratore.

VERO FALSO



- 1 Un byte serve per rappresentare una lettera, una cifra o un simbolo.
- 2 Con un byte si possono ottenere fino a 128 combinazioni di 0 e 1.
- 3 Un diagramma di flusso può avere più di un blocco **Inizio**.
- 4 Un elaboratore è in grado di operare solo su dati digitali.

V	F
V	F
V	F
V	F

RISPOSTA SINGOLA (○) O MULTIPLA (□)



- 1 Il programma compilatore:
 - è sinonimo di programma interprete
 - genera un file con estensione **.acddb**
 - non genera una copia del programma in linguaggio macchina
 - traduce l'intero programma
- 2 Il linguaggio di programmazione:
 - è un linguaggio intermedio tra il linguaggio macchina e il linguaggio naturale
 - consente la scrittura di un programma a prescindere dall'elaboratore in cui sarà eseguito
 - è sinonimo di linguaggio naturale
 - può dare adito ad ambiguità
- 3 L'algoritmo è:
 - indeterministico
 - specifico
 - generale
 - finito
- 4 Indica l'affermazione corretta:
 - Assembler è un linguaggio di programmazione di alto livello
 - un programma scritto in linguaggio di alto livello può funzionare su qualsiasi elaboratore
 - il sistema di numerazione binario non è posizionale
 - l'estensione **.exe** identifica un file di immagine