

Flavia Lughezzani Daniela Princivalle



ST@RTUP

Corso di informatica
per il primo biennio

1

APPRENDERE, COOPERARE, SPERIMENTARE

Edizione **OPENSCHOOL**

1	LIBRODITESTO
2	E-BOOK+
3	RISORSEONLINE
4	PIATTAFORMA

HOEPLI

FLAVIA LUGHEZZANI DANIELA PRINCIVALLE

St@rt Up

APPRENDERE, COOPERARE, SPERIMENTARE

Informatica per il primo biennio

Volume 1



EDITORE ULRICO HOEPLI MILANO

Copyright © Ulrico Hoepli Editore S.p.A. 2019

Via Hoepli 5, 20121 Milano (Italy)

tel. +39 02 864871 – fax +39 02 8052886

e-mail hoepli@hoepli.it

www.hoepli.it



Tutti i diritti sono riservati a norma di legge
e a norma delle convenzioni internazionali

INDICE



MACROAREA **A** CONCETTI BASE E CLOUD

A.1 FONDAMENTI DI INFORMATICA

1 INTRODUZIONE ALL'INFORMATICA

Concetto di algoritmo	2
Diagrammi di flusso	3
• Significato dei blocchi dei diagrammi	3
I connettivi logici AND e OR	4
Il linguaggio naturale e i linguaggi di programmazione	5
I linguaggi di programmazione di alto livello e il linguaggio macchina	6
INFO PAGE Il pensiero computazionale	7
I sistemi di numerazione decimale, binario, esadecimale	9
• Convertire un numero binario in decimale	13
• Convertire un numero decimale in binario	14
• Convertire un numero decimale in esadecimale e viceversa	14
Concetto di bit e di byte	15
Rappresentazione digitale dei dati	15
PALESTRA DELLE COMPETENZE	17

AREA DIGITALE

- La negazione logica NOT
- Utilizzo di pseudo linguaggi

2 HARDWARE E SOFTWARE

Tipi di computer	20
Struttura hardware di un elaboratore	21
CPU: unità centrale di elaborazione	21
Memoria centrale: RAM e ROM	22
INFO PAGE Prestazioni di un computer	22
• Unità di misura della memoria di un computer	23
Porte di input e di output	24
Memorie di massa	24
Periferiche di input	25

Periferiche di output	25
Il sistema operativo	26
Il software applicativo	27
La legalità del software	27
• EULA: il contratto con l'utente finale	28
• Software libero e software proprietario	28
DEBATE Software libero e software proprietario!	29
PALESTRA DELLE COMPETENZE	30

AREA DIGITALE

- Elementi della tastiera
- Sistemi operativi per dispositivi mobili
- Software libero e software proprietario (*approfondimento*)

3 AMBIENTE OPERATIVO

Introduzione a Windows 10	33
Il desktop	34
• Personalizzare il desktop	34
• Opzioni del pulsante Start e personalizzazioni	35
• Creare desktop virtuali	36
Le finestre	37
Gestire file e cartelle	38
• Visualizzare e ordinare elementi	39
• Creare elementi e intervenire su file e cartelle	40
Pannello di controllo e programmi di utilità	41
• Comprimere/decomprimere file e cartelle	41
• Gestione attività	41
• Conoscere le informazioni di base del PC	42
• Creare un Gruppo Home	42
• Aggiungere dispositivi e impostare la stampante predefinita	42
• Disinstallare un programma	43
• Backup dei dati e ripristino del sistema	43
• Deframmentazione e pulizia disco	44
PALESTRA DELLE COMPETENZE	45

AREA DIGITALE

- Creare e gestire account
- Ricercare elementi in specifiche unità o cartelle
- Utilizzare le Raccolte
- Catturare l'immagine dello schermo o della finestra attiva e usare l'app Strumento di cattura
- Altri tipi di backup

A.2 RETI E SICUREZZA

1 RETI, INTERNET E SERVIZI	48
Reti informatiche	49
• Topologia delle reti	49
• Architettura client/server e peer to peer (P2P)	50
• Estensione geografica delle reti	50
• Intranet e rete privata virtuale (VPN)	51
Internet	51
• Connettersi a Internet	52
• Navigare in Internet ed esplorare il Web	53
• L'URL di una risorsa	53
I principali servizi Internet	53
Il Web e alcune sue applicazioni	54
• I social network	54
INFO PAGE Netiquette e privacy	56
• I blog	57
• I servizi VoIP	57
• I forum	57
• I webinar	58
• I wiki	58
• I podcast	58
INFO PAGE L'ergonomia	59
INFO PAGE Studiare e lavorare via Internet	60
DEBATE Social network!	62
PALESTRA DELLE COMPETENZE	63

AREA DIGITALE

- Domini di primo livello
- Dal Web 1.0 al Web 3.0 (*approfondimento*)

2 SICUREZZA INFORMATICA	66
Sicurezza dei dati	66
• Username e password	66
• Il firewall	67
• Il backup	67
• Regole per proteggersi da virus e altri malware	68
Minacce alla sicurezza	68
• I virus e altri malware	68
• Il phishing	69
Sicurezza nelle transazioni online	70

- La crittografia
- Il protocollo https
- Il certificato digitale
- La firma digitale
- Meccanismi di autenticazione

PALESTRA DELLE COMPETENZE	72
----------------------------------	----

AREA DIGITALE

- Rischi nell'uso di strumenti online
- Quarantena ed eliminazione di file infetti

A.3 NAVIGARE, RICERCARE E COMUNICARE

1 USARE IL BROWSER E RICERCARE	75
Creare l'account e accedere a Google Chrome	76
Ambiente Google Chrome	76
Usare e personalizzare Chrome	77
• Impostazioni	78
• Gestire i Preferiti	79
INFO PAGE Symbaloo	80
Operare in Chrome	82
Verificare l'attendibilità delle fonti	82
INFO PAGE Altri browser	84
Il motore di ricerca Google	86
• Avviare una ricerca mirata: per contenuti e usando i filtri	86
• Ricercare in modo avanzato	87
• Ricercare solo in un sito, per tipo di file, utilizzando un'immagine	88
EXTENSION PAGE Google Dictionary	90
• Ricercare notizie con la app Google News	91
• Ricercare libri con la app Google Libri	92
PALESTRA DELLE COMPETENZE	93
2 UTILIZZARE LA POSTA ELETTRONICA	96
Struttura di un indirizzo di posta elettronica	97
Trasmissione di una e-mail	97
La PEC	97
Norme da seguire per la scrittura di una e-mail	98
Ambiente Gmail	98
• Scrivere, revisionare, inviare una e-mail con link e allegati	99
• Inviare e-mail riservate	100
• Memorizzare contatti	101
• Creare gruppi di contatti	101
• Creare cartelle personalizzate	102
• Salvare allegati, rispondere e inoltrare e-mail, stamparle	102

• Operare in modo veloce sui messaggi	103
DEBATE Pro e contro la tecnologia!	104
EXTENSION PAGE Simple Gmail Notes	105
EXTENSION PAGE Boomerang for Gmail	106
PALESTRA DELLE COMPETENZE	107

AREA DIGITALE

- Elementi costitutivi di una mail

A.4 CONDIVIDERE E COLLABORARE

1 LAVORARE IN TEAM	110
Il cloud computing	110
Produrre, collaborare e archiviare con Google Drive	111
• Creare e caricare elementi	112
• Ambienti Documenti, Fogli, Presentazioni	112
• Condividere e gestire elementi	114
INFO PAGE Archiviare e condividere con Dropbox	116
• Inserire commenti e avviare chat	118
• Condivisi con me	119
• Eseguire download di elementi	120



Creare questionari online	120
• Inserire domande e generare il foglio risposte	121
• Impostare il questionario e trasformarlo in quiz	123
• Somministrare il questionario	124
EXTENSION PAGE formLimiter	126
PALESTRA DELLE COMPETENZE	127

AREA DIGITALE

- Vantaggi dell'uso di strumenti e applicazioni online
- Memorie di massa online: vantaggi e limiti
- Applicazioni di produttività via Web
- Caratteristiche delle applicazioni di produttività via Web

2 CONFRONTARSI CON GOOGLE GRUPPI	130
Ambiente Google Gruppi	130
• Creare un gruppo	131
• Impostare categorie o tag	133
• Creare un nuovo argomento e avviare una discussione	133
• Partecipare a una discussione	134
INFO PAGE Le chat con Skype e Hangouts	135
PALESTRA DELLE COMPETENZE	136

MACROAREA **B** OFFICE AUTOMATION

B.1 WORD PROCESSOR

INFO PAGE Word processor	140
1 ELEMENTI BASE DI UN DOCUMENTO	142
Creare un documento	142
Selezionare il testo	143
Copiare e spostare testo	143
Elaborare il testo	143
• La pagina	144
• Il carattere	144
• Il paragrafo	144
• Gli elenchi puntati e numerati	145
• I bordi e gli sfondi	145
Controllare il documento	145
Stampare un documento	145
FISSO I CONCETTI CHIAVE	146
OPERARE CON... Word	147
OPERARE CON... Writer	150

PALESTRA DELLE COMPETENZE 153

AREA DIGITALE

- Nozioni base per una corretta videoscrittura
- Regole per inserire la punteggiatura
- Buoni esempi di allineamento del testo
- Modalità corrette per spaziare i paragrafi
- Personalizzare elenchi puntati e numerati

2 ARRICCHIRE IL TESTO CON LA GRAFICA	163
Gli oggetti grafici	163
FISSO I CONCETTI CHIAVE	164
OPERARE CON... Word	165
OPERARE CON... Writer	168
PALESTRA DELLE COMPETENZE	170

AREA DIGITALE

- Creare immagini d'effetto (Word)
- Applicare una filigrana al documento (Word)
- Creare immagini d'effetto (Writer)
- Applicare una filigrana al documento (Writer)

B.2 FOGLIO ELETTRONICO

INFO PAGE Foglio elettronico	180
1 OPERARE CON IL FOGLIO DI LAVORO	182
In ogni cella un solo dato	183
Inserire dati in una cella	183
Spostare e copiare dati	184
• Usare lo strumento di riempimento automatico	184
Operare su righe e colonne	185
Ordinare i dati	185
FISSO I CONCETTI CHIAVE	186
OPERARE CON... Excel	187
OPERARE CON... Calc	188
PALESTRA DELLE COMPETENZE	189

2 ESEGUIRE CALCOLI ED ELABORARE TABELLE	195
Inserire formule aritmetiche ed espressioni	196
Comprendere l'uso dei riferimenti relativi e assoluti	197
Utilizzare le funzioni base	199
• Le funzioni matematiche	200
• Le funzioni statistiche	200
• Le funzioni di testo	201
Riconoscere e capire i messaggi di errore	202
Elaborare tabelle	202
• I formati numerici	202
• Allineare, orientare i dati, unire più celle	202
FISSO I CONCETTI CHIAVE	203
OPERARE CON... Excel	204
OPERARE CON... Calc	205
PALESTRA DELLE COMPETENZE	206

AREA DIGITALE

- Il riferimento misto
- Applicazioni della funzione ARROTONDA
- I messaggi d'errore

- I formati Contabilità, Numero e Valuta
- Gli Stili cella

3 CREARE GRAFICI E STAMPARE	222
Scegliere il tipo di grafico adatto	223
Creare e personalizzare grafici	224
• Elementi presenti in un grafico	224
Preparazione della stampa	225
Stampare un foglio di lavoro	225
FISSO I CONCETTI CHIAVE	226
OPERARE CON... Excel	227
OPERARE CON... Calc	229
PALESTRA DELLE COMPETENZE	231

AREA DIGITALE

- Miglioramenti per i grafici
- Modificare le voci della legenda

B.3 PRESENTAZIONI

INFO PAGE Presentazioni	246
1 CREARE PRESENTAZIONI E STAMPARE	248
Scelta del tema del layout e degli oggetti	249
L'importanza del titolo delle diapositive	249
FISSO I CONCETTI CHIAVE	250
OPERARE CON... PowerPoint	251
OPERARE CON... Impress	255
PALESTRA DELLE COMPETENZE	259
2 REALIZZARE IPERTESTI E IPERMEDIA	267
Iper testi e ipermedia	268
Costruire una mappa concettuale e lo storyboard	268
Animare una presentazione	269
FISSO I CONCETTI CHIAVE	270
OPERARE CON... PowerPoint	271
OPERARE CON... Impress	272
PALESTRA DELLE COMPETENZE	273



APPENDICE

COMPITI DI REALTÀ • IN PRATICA... FARE E SAPER FARE

Compito 1 LA MIA SCUOLA!	280	Compito 4 SIAMO SPORTIVI?	292
Compito 2 PERCHÉ BIOLOGICO?	284	Compito 5 VISITIAMO BORGHI!	296
Compito 3 IO... ONLINE	288		

PRESENTAZIONE

La disciplina Informatica integra teoria e pratica tecnologica, aspetti concettuali e aspetti applicativi. Queste due linee “parallele e convergenti” definiscono gli obiettivi di questo progetto didattico: trasmettere i principali fondamenti teorici delle scienze dell’informazione e, in parallelo, sollecitare la padronanza di strumenti e ambienti volti alla soluzione di problemi autentici al fine di sviluppare nello studente **competenze esperte**.

ST@RT UP – APPRENDERE, COOPERARE, SPERIMENTARE è un nuovo corso, strutturato in due volumi, destinato agli studenti del **primo biennio** dei **diversi indirizzi scolastici** per la disciplina **Informatica**.

I due volumi, ai quali si può abbinare anche il testo **ST@RT UP – PROGRAMMAZIONE A BLOCCHI**, sono sviluppati nel rispetto delle Linee guida ministeriali e introducono lo studente all’informatica, con una particolare attenzione verso lo sviluppo, oltre che di competenze disciplinari, anche di *soft skills* importanti per qualsiasi contesto di studio e di lavoro.

Questo progetto didattico mira a sollecitare negli alunni un approccio alla conoscenza affine alle loro quotidiane pratiche sociali e a **favorire l’abitudine alla collaborazione, alla progettazione, al confronto, al lavoro in gruppo**, nonché ad **accrescere le abilità di ragionamento e logica**.

I **due volumi** sono **strutturati ognuno in due macroaree** che, combinandosi assieme, vanno a coprire gli aspetti di base dell’informatica, quelli degli applicativi di Office Automation e quelli degli ambienti e strumenti del Web 2.0 e del *cloud computing*. Ogni volume, inoltre, è corredato da un’**appendice innovativa**, che propone un insieme di **compiti di realtà** appositamente pensati per avvicinare gli studenti all’approccio pedagogico della *flipped classroom* e per accompagnarli nei percorsi per le competenze trasversali e per l’orientamento. Una particolare attenzione è poi rivolta anche alla **didattica inclusiva**, con mappe di inizio unità, mappe da completare e numerosi esercizi e schede esercitative proposti nel font ad alta leggibilità.

Lo studente è coinvolto come parte attiva e allenato alla logica, all’ordine procedurale, alla correttezza espositiva, alla sicurezza operativa e a un metodo di lavoro autonomo ed efficace.

Contenuti del volume

Macroarea A – CONCETTI BASE E CLOUD

In questa macroarea vengono poste le basi dell’informatica: le definizioni principali, l’**architettura del computer**, le caratteristiche tecniche e le funzionalità dei suoi componenti hardware e software. Si illustrano le proprietà e le funzionalità di base del **sistema operativo Windows 10** e viene fornita una base teorica che consente di utilizzare in maniera appropriata la Rete, con riferimenti alle problematiche legali relative al **diritto d’autore** e alla **privacy**. Sono inoltre presentati i **principali servizi Internet** e descritti gli aspetti essenziali della **sicurezza informatica**.

Sono infine illustrati numerosi **ambienti cloud**, con un approccio volto a servirse-ne quali strumenti per strutturare e presentare contenuti, ma anche per favorire la collaborazione, il confronto, il lavoro di gruppo, la progettazione: l’**impostazione**



spiccatamente **operativa** di questi moduli è orientata a un corretto e consapevole utilizzo della Rete nella **ricerca di informazioni**, nella **comunicazione**, per l'**accesso ai servizi** e per **prelevare materiale dal Web**.



Macroarea B – OFFICE AUTOMATION

I contenuti di questa macroarea forniscono conoscenze e tecniche di base per un utilizzo appropriato e razionale degli strumenti di cui dispongono gli applicativi dei pacchetti **Office 2016** e **LibreOffice 6.1** (**Word processor**, **Foglio elettronico** e **Presentazioni**), intesi anche come strumenti trasversali per la risoluzione di problemi interdisciplinari. Gli esercizi proposti guidano lo studente a una graduale acquisizione di tecniche per realizzare documenti accattivanti, eseguire semplici calcoli, elaborare tabelle, creare grafici, stampare fogli di lavoro, nonché realizzare presentazioni ipertestuali e multimediali per una comunicazione chiara ed efficace. Specifici esercizi propongono un proficuo utilizzo dei software e allenano a individuare soluzioni adeguate alle situazioni proposte.



Appendice – COMPITI DI REALTÀ

Conclude il volume una innovativa **Appendice** che propone una serie di **esercizi calati in contesti reali**, organizzati secondo la **metodologia della flipped classroom** e strutturati per essere svolti in gruppo. Nell'Appendice ogni compito mobilita le conoscenze e le abilità acquisite nelle macroaree, combinandole assieme al fine di sviluppare e consolidare competenze disciplinari e di cittadinanza.

Apparato didattico ed esercitativo

Le due **macroaree** del volume si articolano in **moduli** tra loro indipendenti che, a loro volta, si compongono di diverse **unità di apprendimento**.

Ogni unità utilizza **icone**, **schemi**, **rubriche** e **impostazioni standard** che guidano lo studente all'**acquisizione di conoscenze** e allo **sviluppo di abilità**.


Le conoscenze, la padronanza terminologica specifica, le abilità e le competenze raggiunte sono irrobustite in diverse unità mediante la proposta di **test**, **schede organizzative**, **casi**, **attività di gruppo**, **compiti di realtà**. Le diverse attività sono tutte volte a rafforzare la capacità di ragionamento e di logica, a sviluppare e consolidare competenze digitali, di cittadinanza e *soft skills*.

Nella **versione digitale** del volume (**eBook+**) sono inoltre disponibili **approfondimenti tematici** e **test interattivi**.

ECDL

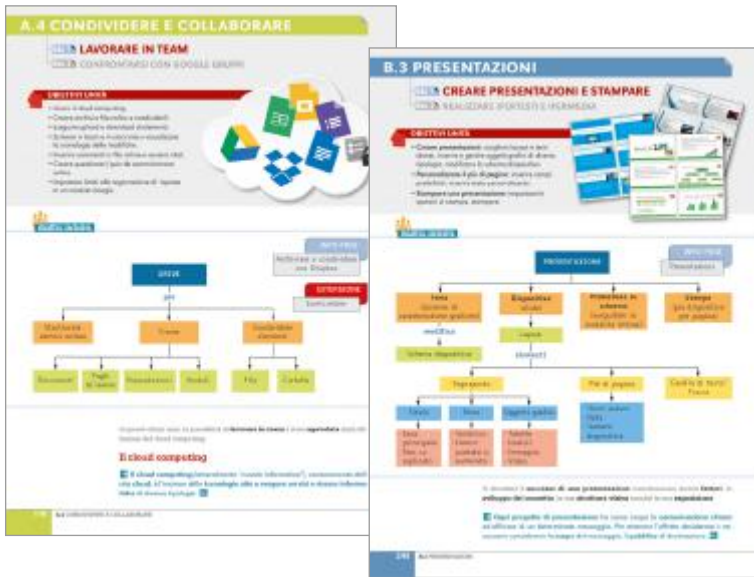
I contenuti proposti includono conoscenze e tecniche utili a una **preparazione informatica** non specialistica, ma comunque tale da consentire l'apprendimento di numerosi argomenti riguardanti gli esami **per il conseguimento** della certificazione informatica prevista dalla **Patente Europea del Computer** (ECDL).

Materiali online

I **file** da aprire richiamati negli **Esercizi** e nelle schede della sezione **Palestra delle competenze** della **Macroarea B** (archiviati nelle cartelle **B1 Word processor**, **B2 Foglio elettronico** e **B3 Presentazioni**) sono scaricabili da www.hoepliscuola.it ( hoepliscuola.it), unitamente a numerosi **approfondimenti tematici** (fruibili anche dall'eBook+).

STRUTTURA DEL CORSO PER IMMAGINI

APPARATO DIDATTICO



MAPPA CONCETTUALE

In apertura di ogni unità, la mappa concettuale offre una sintetica anticipazione dei contenuti sviluppati, fungendo al contempo da schema riepilogativo e di sistematizzazione dei saperi per abituare lo studente all'apprendimento e alla sintesi dei contenuti attraverso un percorso ragionato.



INFO PAGE

Schede con approfondimenti, *focus* su ambienti e confronti delle interfacce degli applicativi o ambienti.

TEST AL VOLO
 Esercizi flash presenti in alcune unità delle due macroaree, finalizzati a richiamare velocemente l'attenzione dello studente su alcuni dei contenuti proposti per rinforzarli.



CURIOSITÀ

Box di consigli, precisazioni o curiosità sugli argomenti sviluppati nei paragrafi e approfondimenti su alcuni dei contenuti trattati.

DEBATE
 Scheda mediante la quale viene proposta la metodologia didattica *Debate*, volta a sviluppare competenze trasversali quali il saper collaborare, negoziare, confrontarsi, dialogare per esercitare consapevolmente un ruolo attivo nei processi decisionali.

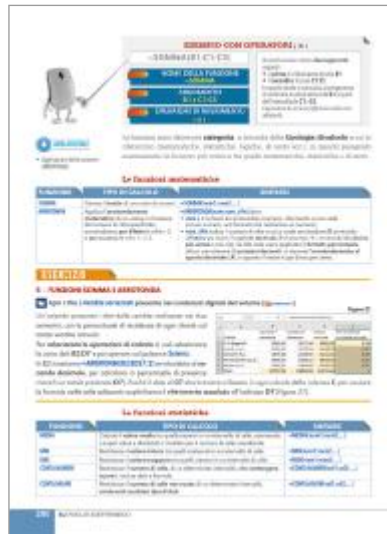


EXTENSION PAGE

Schede che illustrano diverse estensioni che potenziano alcuni degli strumenti Web proposti, arricchendoli con ulteriori funzionalità.

ESERCIZI/CASI GUIDATI/ESEMPI

All'interno di alcune unità, accompagnano lo studente nell'apprendimento dei concetti teorici permettendogli di meglio comprendere i contenuti proposti (*sapere*) per acquisire così le abilità (*saper fare*) necessarie per affrontare con più sicurezza gli esercizi proposti nelle schede della sezione "Palestra delle competenze".



RIFLETTO... e ORGANIZZO...
Schede presenti in alcune unità della Macroarea A finalizzate ad attività di metacognizione per avviare lo studente ad acquisire un metodo di studio efficace.

FISSO I CONCETTI CHIAVE

Scheda presente nella Macroarea B, collocata dopo la spiegazione teorica con lo scopo di fissare le conoscenze acquisite, rendendo l'alunno consapevole dei contenuti appresi prima di affrontare l'aspetto operativo.



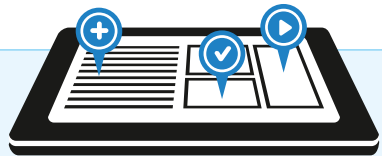
A DOMANDA... RISPONDO!
Scheda presente al termine di alcune unità della Macroarea A e in tutte quelle della Macroarea B: attraverso test di diversa tipologia, consente di verificare l'acquisizione dei contenuti (*sapere*). Nella versione digitale (eBook+) questi test sono interattivi e autocorrettivi.



RISOLVO IL CASO...
Scheda presente al termine di alcune unità della Macroarea A, che propone situazioni scolastiche di lavoro individuale e di gruppo allo scopo di allenare lo studente a servirsi degli ambienti proposti per un uso personale e collaborativo.



AREA DIGITALE (eBook+)



APPROFONDIMENTI ONLINE
 Schede con approfondimenti sugli argomenti trattati e procedure operative per un utilizzo corretto degli applicativi proposti.

ESERCIZI INTERATTIVI
 Nella versione digitale del volume i test a risposta chiusa della scheda "A domanda... rispondo!" sono interattivi e autocorrettivi.



MOMENTI DI CONFRONTO..., E ORA PROVO IO!, PROGETTO E CREO!
 Schede proposte nella Macroarea B al termine di ogni unità, con esercizi di varia tipologia e distinti per livello di difficoltà. Gli esercizi della scheda "Momenti di confronto...", mediante compiti autentici (casi), stimolano la collaborazione tra studenti e permettono di esaminare i problemi da diverse prospettive teoriche e pratiche, favorendo l'individuazione di soluzioni alternative che sollecitano la creatività personale e la cooperazione di gruppo.

COOPERANDO... SI IMPARA!
 Scheda presente in alcune unità della Macroarea A, che propone attività da svolgere in gruppo, per avviare gli studenti all'approccio didattico della *flipped classroom*.



APPENDICE "COMPITI DI REALTÀ"
 Sezione che chiude il volume proponendo esercizi calati in contesti reali, strutturati secondo la metodologia della *flipped classroom*. In ogni compito di realtà proposto trovano riferimento le competenze chiave sollecitate, così come definite dal MIUR e dal Consiglio europeo.

L'OFFERTA DIDATTICA HOEPLI

L'edizione **Openschool** Hoepli offre a docenti e studenti tutte le potenzialità di Openschool Network (ON), il nuovo sistema integrato di contenuti e servizi per l'apprendimento.

Edizione **OPENSCHOOL**



LIBRO DI TESTO



Il libro di testo è l'**elemento cardine** dell'offerta formativa, uno strumento didattico **agile e completo**, utilizzabile **autonomamente** o in combinazione con il ricco **corredo digitale** offline e online. Secondo le più recenti indicazioni ministeriali, volume cartaceo e apparati digitali **sono integrati in un unico percorso didattico**. Le espansioni accessibili attraverso l'eBook+ e i materiali integrativi disponibili nel sito dell'editore sono puntualmente richiamati nel testo tramite apposite icone.



eBOOK+



L'**eBook+** è la versione digitale e interattiva del libro di testo, utilizzabile su **tablet, LIM e computer**. Aiuta a comprendere e ad approfondire i contenuti, rendendo l'apprendimento più attivo e coinvolgente. Consente di leggere, annotare, sottolineare, effettuare ricerche e accedere direttamente alle numerose **risorse digitali integrative**.
→ Scaricare l'eBook+ è molto **semplice**. È sufficiente seguire le istruzioni riportate nell'ultima pagina di questo volume.



RISORSE ONLINE



Il sito della casa editrice offre una ricca dotazione di **risorse digitali** per l'approfondimento e l'aggiornamento. Nella pagina web dedicata al testo è disponibile **MyBookBox**, il contenitore virtuale che raccoglie i materiali integrativi che accompagnano l'opera.
→ Per accedere ai materiali è sufficiente registrarsi al sito **www.hoepliscuola.it** e inserire il codice coupon che si trova nella terza pagina di copertina. **Per il docente** nel sito sono previste ulteriori risorse didattiche dedicate.



PIATTAFORMA DIDATTICA



La **piattaforma didattica** è un ambiente digitale che può essere utilizzato in modo duttile, a misura delle esigenze della classe e degli studenti. Permette in particolare di **condividere contenuti** ed **esercizi** e di partecipare a **classi virtuali**. Ogni attività svolta viene salvata sul **cloud** e rimane sempre disponibile e aggiornata. La piattaforma consente inoltre di consultare la versione online degli eBook+ presenti nella propria libreria.
→ È possibile accedere alla piattaforma attraverso il sito **www.hoepliscuola.it**.

CONCETTI BASE E CLOUD



A.1 **FONDAMENTI DI INFORMATICA**

- 1 Introduzione all'informatica
- 2 Hardware e software
- 3 Ambiente operativo

A.2 **RETI E SICUREZZA**

- 1 Reti, internet e servizi
- 2 Sicurezza informatica

A.3 **NAVIGARE, RICERCARE E COMUNICARE**

- 1 Usare il browser e ricercare
- 2 Utilizzare la posta elettronica

A.4 **CONDIVIDERE E COLLABORARE**

- 1 Lavorare in team
- 2 Confrontarsi con Google gruppi

A.1 FONDAMENTI DI INFORMATICA

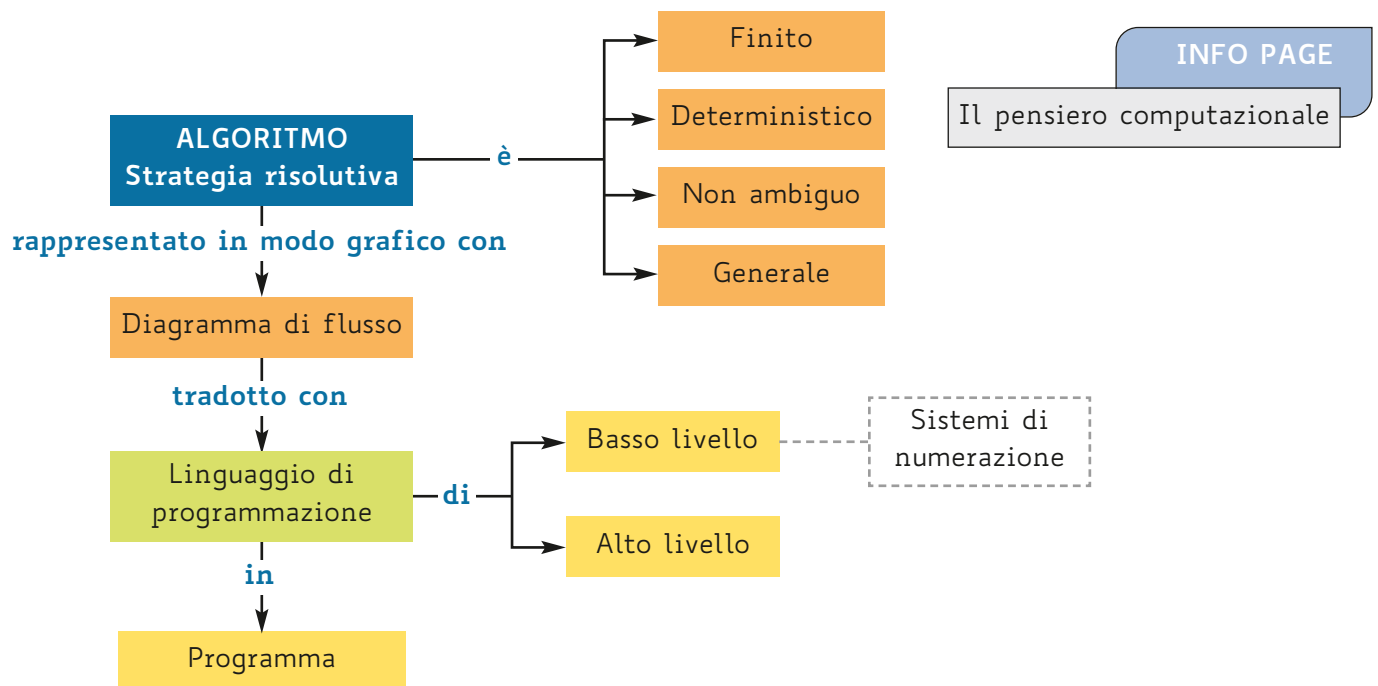
Uda 1 INTRODUZIONE ALL'INFORMATICA

Uda 2 HARDWARE E SOFTWARE

Uda 3 AMBIENTE OPERATIVO

OBIETTIVI UNITÀ

- Conoscere i concetti di algoritmo, diagramma di flusso, programma.
- Distinguere i diversi linguaggi di programmazione.
- Elencare i sistemi di numerazione binario, decimale ed esadecimale.
- Conoscere come avviene la rappresentazione digitale dei dati.



CURIOSITÀ

L'informatica (dal francese *Information Automatique*) è la scienza che si occupa dei sistemi per elaborare e gestire in modo automatico le informazioni.

Tutte le volte in cui si deve risolvere un problema o soddisfare una necessità occorre:

- individuare gli elementi che si hanno a disposizione (**dati di input**);
- stabilire l'obiettivo che si vuole raggiungere (**dati di output**);
- precisare la **procedura** che si intende seguire per ottenere l'**obiettivo** prefissato.

Tante sono le situazioni che, nella vita quotidiana, richiedono l'esecuzione di una **procedura**, cioè il compimento di una **sequenza precisa, ordinata e finita di**

azioni che conducono all'obiettivo fissato. Tale sequenza, in informatica, viene descritta attraverso l'**algoritmo**.

Per comprendere meglio questo concetto, esaminiamo l'esempio di una famiglia che ha acquistato la macchina per fare il pane (*Figura 1*). Avremo, in questo caso:

- come **input**: farina, lievito, acqua, sale, olio;
- come **algoritmo risolutivo**: programma di impasto e cottura scelto (istruzioni che conducono al risultato fissato);
- come **output**: pane.



L'esempio può essere trasferito in ambito informatico e può aiutarci a comprendere l'attività svolta da un PC per soddisfare le nostre esigenze quotidiane (*Figura 2*).



Concetto di algoritmo

⇒ L'algoritmo è la strategia risolutiva di un problema, costituito da una sequenza finita di operazioni (dette anche passi o istruzioni) che consente di risolvere tutti i problemi di una determinata "classe" e produrre il risultato stabilito. ←

L'algoritmo, in quanto tale, deve essere:

- **finito**, ovvero essere costituito da un numero finito di passi eseguiti un numero finito di volte;
- **deterministico**, cioè, a partire dagli stessi dati di input, deve produrre i medesimi risultati;
- **non ambiguo**, in quanto le operazioni che lo costituiscono devono poter essere interpretate in modo univoco anche da esecutori diversi;
- **generale**, ovvero deve fornire la soluzione per tutti i problemi che appartengono alla medesima classe.

Diagrammi di flusso

Per descrivere in **forma algoritmica** la **procedura risolutiva** di un problema è necessario rendere visivamente comprensibile la **successione logica** delle **istruzioni** di cui si compone l'algoritmo procedendo con la sua **rappresentazione grafica** me-

CURIOSITÀ

I dati che interessano l'algoritmo possono essere:

- **costanti**, quando nello svolgimento dell'algoritmo mantengono sempre lo stesso valore (si tratta in genere di numeri);
- **variabili**, quando durante l'elaborazione possono assumere valori diversi (vengono in generale rappresentati da lettere, a ognuna delle quali sarà assegnata una determinata variabile).

diante la costruzione di **diagrammi di flusso**, definiti anche diagrammi a blocchi, o *flow-charts*. Tali diagrammi costituiscono lo strumento per **presentare il ragionamento** che condurrà al risultato atteso.

I **blocchi** di cui si compongono sono figure geometriche nelle quali sono inserite le **operazioni elementari**. Tali blocchi sono collegati tra loro da linee di giunzione, orientate e disposte per stabilire la sequenza in cui si desidera che le operazioni vengano svolte, con possibilità di collegamenti diversi per risolvere le molteplici situazioni che si presentano e che dovranno essere soddisfatte dall'algoritmo.

In algoritmi realizzati per la soluzione di problemi matematici, le istruzioni si sviluppano mediante **costanti e variabili** legate da **operatori aritmetici**, per esempio +, -, * ecc., il cui risultato sarà assegnato a una variabile.




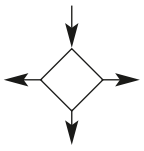



ESEMPIO 1

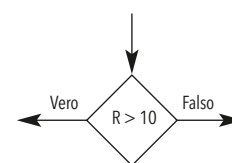
Di seguito mostriamo un esempio di istruzioni espresse con costanti e variabili.

$$A = B - 10 \text{ significa}$$

“sottrai al valore della **variabile B** il valore della **costante 10** e **assegna** il risultato alla **variabile A**”.

Significato dei blocchi dei diagrammi

BLOCCO	SIGNIFICATO
Inizio 	Serve per indicare il punto di partenza dell'algoritmo e al suo interno possono apparire le scritte INIZIO o START : in un diagramma di flusso ne è presente solo uno.
Ingresso/ Uscita 	Viene utilizzato per le operazioni di ingresso o scrittura dei dati (input) e di uscita o lettura dei risultati (output). Ha una sola linea di giunzione in entrata e in uscita.
Elaborazione/ Azione 	Serve per le operazioni generiche (operazioni di calcolo) o di assegnazione e al suo interno viene descritta in forma simbolica o letterale l' operazione da eseguire . Ha una sola linea di giunzione in entrata e in uscita.
Decisione/ Domanda/ Confronto 	<p>Serve per le operazioni di confronto, controllo e di scelta tra alternative per percorrere sequenze di istruzioni diverse in funzione del verificarsi o meno di certe situazioni; può essere a un solo ingresso e a due o più uscite. Al suo interno si inserisce l'espressione da considerare, il cui risultato viene confrontato con variabili e costanti attraverso gli operatori = uguale, > maggiore, >= maggiore o uguale, < minore, <= minore o uguale, <> diverso, V (vero), F (falso), questi ultimi due indicati sulle linee di giunzione (in questo caso definite di uscita).</p> <p>Nell'esempio proposto si dovrà procedere con le operazioni in un senso oppure nell'altro, in base al fatto che la condizione posta risulti vera oppure falsa.</p>
Linee di giunzione/ Direzione del flusso 	La freccia posta al termine di ogni blocco o segmento collega al passo logico successivo da prendere in considerazione e quindi indica il flusso di esecuzione delle operazioni.
Fine 	Serve per indicare la fine dell'algoritmo e nel suo interno possono apparire le scritte FINE o STOP . In genere in un algoritmo ne è presente uno solo.
Connettori 	Individuano dei punti dai quali si può fare riferimento ad altre parti. Per esempio, il connettore ① —> trasferisce il proseguimento del diagramma dal punto in cui si trova al punto dove compare il connettore —> ① contenente lo stesso numero.



Gli algoritmi vengono utilizzati per costruire diagrammi, o flow chart, e per la soluzione di problemi di calcolo.

ESEMPIO 2

Consideriamo il seguente problema: dato in input un numero, calcolarne il cubo se è maggiore di 0, il quadrato in caso contrario.

VARIABILI UTILIZZATE

- **NUM**, che rappresenta il numero (reale) in input;
- **RIS**, che rappresenta il risultato (reale) in output.

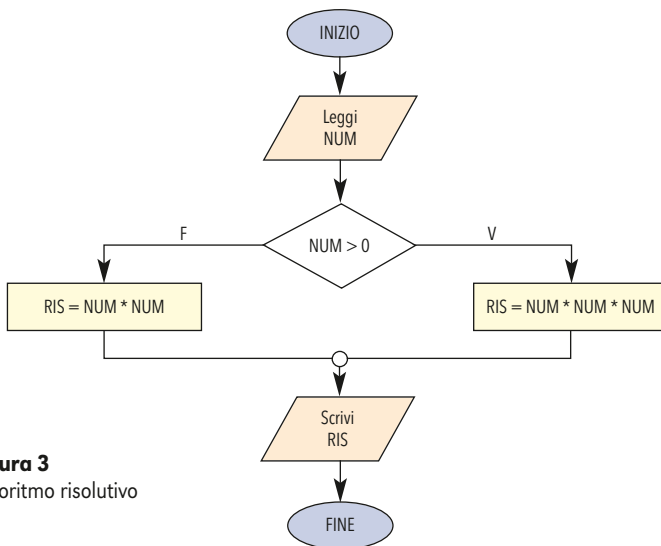


Figura 3
Algoritmo risolutivo

Se osserviamo l'algoritmo risolutivo di *Figura 3*, vediamo che innanzitutto è necessario prendere in input un numero, effettuare il confronto del valore con zero, quindi:

- se il numero è maggiore di zero, dovrà essere moltiplicato tre volte per se stesso;
- se il numero è minore di zero, dovrà essere moltiplicato due volte per se stesso.

Il risultato sarà sicuramente un numero positivo, oppure sarà uguale a zero se il numero di partenza è 0.

I connettivi logici AND e OR

Come si può notare dalla rappresentazione dell'algoritmo, l'elaboratore è programmato per eseguire determinate funzioni che si basano sulla verità o sulla falsità di certe proposizioni semplici, come per esempio "Il Po è un fiume" (proposizione semplice vera), oppure "2 + 2 = 3" (proposizione semplice falsa). La verità o la falsità di una proposizione semplice viene indicata con l'espressione "valore di verità", nel senso che una proposizione può essere **vera** o **falsa**, ma non entrambe le cose.

Le proposizioni semplici possono tuttavia essere combinate tra loro mediante i connettivi logici **AND** e **OR** in modo da dare origine a una nuova proposizione, detta "composta", il cui valore di verità è determinato dal valore di verità delle sue proposizioni semplici e dai connettivi logici che le collegano.

➔ **Un connettivo, o operatore logico, istituisce fra due proposizioni a e b una relazione che dà origine a una terza proposizione c che sarà vera o falsa in base ai valori delle due proposizioni a e b e al tipo di connettivo utilizzato.** ←

A pagina seguente prendiamo in considerazione le proposizioni semplici **a** e **b** collegate ai connettivi logici **AND** e **OR**, mostrando le relative **tavole di verità**.

CURIOSITÀ

I valori **vero** o **falso** sono definiti **valori logici** o **valori booleani**.

AREA DIGITALE

- La negazione logica NOT

Connettivo AND (congiunzione)

Date le proposizioni semplici **a** e **b** collegate dal connettivo **AND**, il risultato della congiunzione **a AND b** darà origine a una proposizione composta che sarà:

- vera, solo se **a** e **b** sono entrambe vere;
- falsa, quando almeno una delle due è falsa.

Tabella di verità della proposizione composta **a AND b**

a	b	a AND b
v	v	v
v	f	f
f	v	f
f	f	f



a (prima proposizione)	Connettivo AND	b (seconda proposizione)	Proposizione composta
Il Po è un fiume	e	$2 + 2 = 4$	Vero
Il Po è un fiume	e	$2 + 2 = 3$	Falso
Il Po è un lago	e	$2 + 2 = 4$	Falso
Il Po è un lago	e	$2 + 2 = 3$	Falso



Connettivo OR (disgiunzione)

Date le proposizioni semplici **a** e **b** collegate dal connettivo **OR**, il risultato della disgiunzione **a OR b** darà origine a una proposizione composta che sarà:

- falsa, solo se **a** e **b** sono entrambe false;
- vera, quando almeno una delle due è vera.

Tabella di verità della proposizione composta **a OR b**

a	b	a OR b
v	v	v
v	f	v
f	v	v
f	f	f



a (prima proposizione)	Connettivo OR	b (seconda proposizione)	Proposizione composta
Il Po è un fiume	oppure	$2 + 2 = 4$	Vero
Il Po è un fiume	oppure	$2 + 2 = 3$	Vero
Il Po è un lago	oppure	$2 + 2 = 4$	Vero
Il Po è un lago	oppure	$2 + 2 = 3$	Falso

Il linguaggio naturale e i linguaggi di programmazione

In generale, quando si parla di **linguaggio**, ci si riferisce:

- al **lessico**, ossia all'insieme delle parole, ciascuna con il suo significato, disponibili per esempio in un dizionario;
- alla **sintassi**, costituita dalle regole da seguire per costruire una frase in modo corretto, affinché possa essere compresa.

Il **linguaggio naturale** è quello comunemente utilizzato per comunicare con gli altri e consente ricchezza espressiva; in alcuni casi, tuttavia, può dare adito ad ambiguità e talvolta anche a ridondanza. Vediamo un esempio.

TEST AL VOLO

Osserva la figura.
Si tratta di un blocco di:

- elaborazione
 ingresso

ESEMPIO 3

Consideriamo la seguente frase: "La bella pesca".

Essa può essere interpretata in due modi:

- 1 "una bella ragazza sta pescando";
- 2 "un bel frutto".

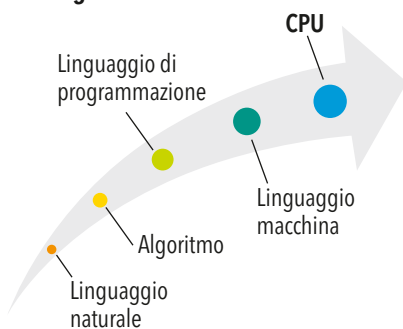
1



2



Figura 4



L'esempio appena esaminato ci fa comprendere che il linguaggio naturale non può essere utilizzato per "istruire" l'elaboratore su ciò che deve compiere, in quanto le caratteristiche che presenta non sono adatte alla sua logica: **un elaboratore non può gestire informazioni ambigue**.

L'**algoritmo**, invece, per le sue caratteristiche, è **idoneo a istruire un elaboratore**. Tuttavia, le istruzioni di cui si compone, devono essere tradotte, mediante un **linguaggio di programmazione** (costituito da un **alfabeto** e da un insieme di **regole sintattiche** per l'uso corretto delle parole del linguaggio), in un **programma**.

Il **linguaggio di programmazione** è un linguaggio **intermedio** fra **linguaggio macchina** (direttamente comprensibile dall'elaboratore) e il linguaggio che descrive gli algoritmi (*Figura 4*).

Esso viene definito anche **linguaggio di alto livello** in quanto è costituito da una sintassi che non è dipendente dal funzionamento specifico di una certa CPU: di conseguenza, consente di scrivere un programma indipendentemente dall'elaboratore su cui sarà eseguito.

I linguaggi di programmazione di alto livello e il linguaggio macchina

I linguaggi di programmazione di alto livello tendono ad assomigliare ai linguaggi naturali proprio per rendere più facile la programmazione del computer. Il computer, invece, capisce solo il **linguaggio macchina (o linguaggio di basso livello)**, direttamente comprensibile dalla CPU, costituito da una sintassi limitatissima e molto rigida, composta da sequenze di **cifre binarie** 1 e 0.

I primi computer si basavano esclusivamente sul linguaggio macchina: il programmatore utilizzava una tabella che serviva per tradurre i comandi nella serie corrispondente di cifre binarie da inserire; era sufficiente sbagliare una cifra per mettere in crisi tutto il sistema. Inoltre il linguaggio di basso livello, direttamente compreso ed eseguito dal processore, è strettamente collegato alla sua struttura fisica, con la conseguenza che il medesimo programma può non funzionare in microprocessori diversi.

Proprio per ovviare a questi inconvenienti sono stati sviluppati i **linguaggi di programmazione di alto livello**, in cui le istruzioni non sono più indicate da sequenze di cifre binarie, ma da nomi simbolici, più facili da riconoscere, memorizzare e utilizzare da parte del programmatore.

Tuttavia, poiché l'elaboratore riesce a interpretare solo istruzioni formulate in linguaggio macchina nelle sequenze di 1 e 0, sarà necessario tradurre i programmi scritti in linguaggio di alto livello in linguaggio macchina.

Questa operazione di traduzione può essere eseguita dai programmi di seguito descritti.

- **Programmi compilatori:** traducono l'intero programma scritto in linguaggio di alto livello (per esempio, in Pascal o in C++) nella corrispondente copia in linguaggio macchina; tutte le istruzioni vengono controllate nel lessico e nella sintassi, tradotte e trasformate in un file eseguibile (con estensione **.exe** nei sistemi Windows) che diventa indipendente dal programma scritto in linguaggio di alto livello, potendo così essere eseguito senza il programma compilatore.

CURIOSITÀ

Le operazioni elementari eseguite dalla CPU sono rappresentate come sequenze di cifre binarie.

Di conseguenza, a questo livello, un programma corrisponde a una serie di **istruzioni espresse in codice binario, direttamente eseguibili dalla CPU**.

CURIOSITÀ

Il linguaggio **Assembler** è un **linguaggio di programmazione di basso livello** che ha la stessa struttura del linguaggio macchina ed è strettamente legato alle caratteristiche del computer per il quale è stato definito. È complicato da memorizzare in quanto vi è una precisa corrispondenza tra le istruzioni simboliche del linguaggio Assembler e le istruzioni definite in linguaggio macchina.

- **Programmi interpreti:** “leggono” riga per riga le istruzioni scritte in linguaggio di alto livello, ne controllano il lessico e la sintassi e le traducono in linguaggio macchina per farle eseguire direttamente dall’unità centrale di elaborazione. In questo caso non viene prodotta una copia del programma in linguaggio macchina, ma ogni istruzione viene di volta in volta tradotta e poi fatta eseguire.

TEST AL VOLO

Il linguaggio macchina è un linguaggio:

- di alto livello
- di basso livello

AREA DIGITALE

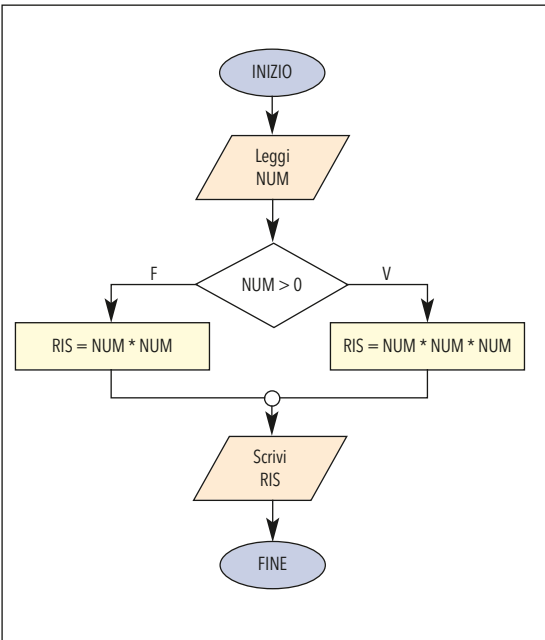
- Utilizzo di pseudo linguaggi

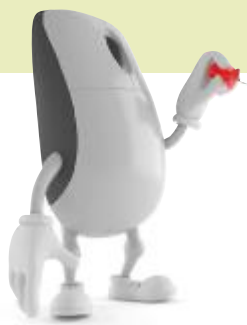
Per rispondere alle diverse esigenze sono stati sviluppati diversi linguaggi di programmazione di alto livello, alcuni dei quali sono indicati di seguito:

- **SCRATCH**, ambiente visuale particolarmente utilizzato in ambito scolastico per progetti pedagogici;
- **PYTHON**, orientato agli oggetti, è adatto per lo sviluppo software;
- **C**, utilizzato per applicazioni di tipo ingegneristico;
- **JAVA**, progettato per applicazioni su Internet;
- **PASCAL**, particolarmente adatto per la didattica della programmazione;
- **BASIC**, inizialmente progettato per applicazioni scientifiche e didattiche, viene attualmente utilizzato per ogni tipo di problema. Grazie alla facilità con cui viene assimilato e alla grande diffusione dei PC, oggi viene usato nella versione **VISUAL BASIC**.

ESEMPIO 4

Di seguito rappresentiamo in linguaggio PASCAL e in linguaggio VISUAL BASIC il diagramma di pag. 4, riproposto qui sotto.

PASCAL	
 <pre> graph TD INIZIO([INIZIO]) --> LeggiNUM[/Leggi NUM/] LeggiNUM --> Decision{NUM > 0} Decision -- F --> RIS1[RIS = NUM * NUM] Decision -- V --> RIS2[RIS = NUM * NUM * NUM] RIS1 --> Merge(()) RIS2 --> Merge Merge --> ScriviRIS[/Scrivi RIS/] ScriviRIS --> FINE([FINE]) </pre>	<pre> program NUMERO; var NUM : Real; RIS : Real; begin Writeln ('DAMMI IL NUMERO'); Readln(NUM); if NUM > 0 then RIS:= NUM * NUM * NUM else RIS:= NUM * NUM; Writeln ('IL RISULTATO E': ', RIS:10:2); end. </pre>
VISUAL BASIC	
<pre> Sub NUMERO() Dim NUM As Single 'numero in input Dim RIS As Single 'risultato NUM = Val(InputBox("Dammi il numero")) If NUM > 0 Then RIS = NUM * NUM * NUM Else RIS = NUM * NUM End If MsgBox ("il risultato è" & Str(RIS)) End Sub </pre>	



info PAGE

Il pensiero computazionale

1

Programma il Futuro

Indirizzo e-mail Password ACCEDI

IL PROGETTO - CHI - PERCORSI - LA COMUNITA - NOTIZIE - AIUTO -

Ultimi aggiornamenti sul concorso Codi-Amo!

Tweet @ProgrammaFuturo

Programma il Futuro

Iscriviti Percorsi didattici Perché

Figura A

Programma il Futuro

1. Fai clic su CHI e scegli Iscrizione per studenti/CLICCA QUI PER ISCRIVERTI per registrarti e poi eseguire l'accesso

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 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**,



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 Il Labirinto per accedere ai diversi percorsi



Figura C
Pagina Lezione

Fornisce gli obiettivi della lezione e spiega le caratteristiche dell'ambiente Il Labirinto

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



Figura E
Menu Corsi



Figura F
Sezione L'Or del Codice in Code.org
Scegli Labirinto Classico.

fai clic su **L'Or del Codice** e poi su **Il Labirinto** (Figura B), attiva quindi il comando **Accedi alla lezione** (Figura C); 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). Nel menu, fai clic su **Corsi** (Figura E) e poi scegli **Labirinto Classico** (Figura F) per iniziare il percorso.

Le attività proposte non sono banali, come potrebbe apparire a un primo approccio, perché abituanò la mente a inventare e poi definire precise procedure per risolvere le diverse situazioni che si presentano, come accade nella quoti-

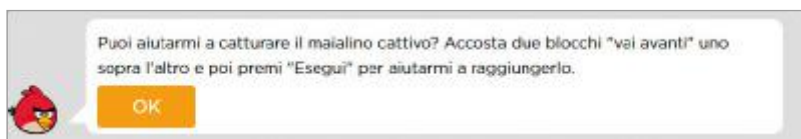


Figura G
Box richiesta Esercizio 1

Figura H
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

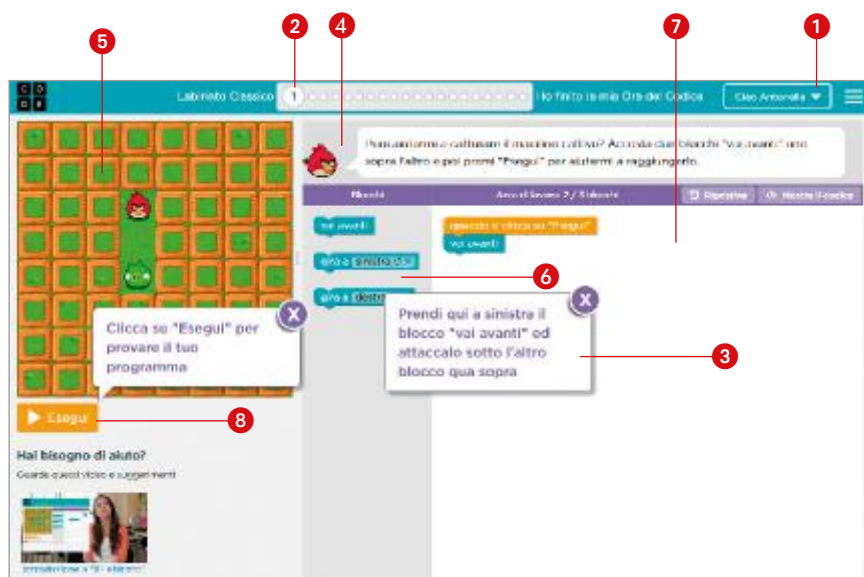


Figura I
Feedback Esercizio 1



Figura J
Esempio di attestato L'Ora del Codice



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/>, fare clic su **Accedi**, inserire le tue credenziali di registrazione e poi confermare con **Accedi**. Nella pagina caricata (Figura K) puoi scegliere se continuare la lezione e proseguire con il successivo esercizio, oppure se visualizzare il corso (Figura L) per avere un'idea del punto in cui ti trovi



Figura K Accesso all'area personale di Code.org

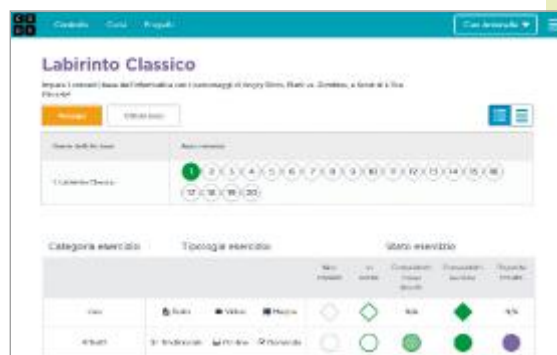


Figura L
Visualizza il corso

dianità, in momenti di vita personale o professionale. Dopo aver visionato il video introduttivo, compare la prima richiesta (Figura G) e, facendo clic su **OK**, l'ambiente in cui costruire la soluzione (Figura H) e verificarla.

Otteni subito un feedback (Figura I) 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 J).

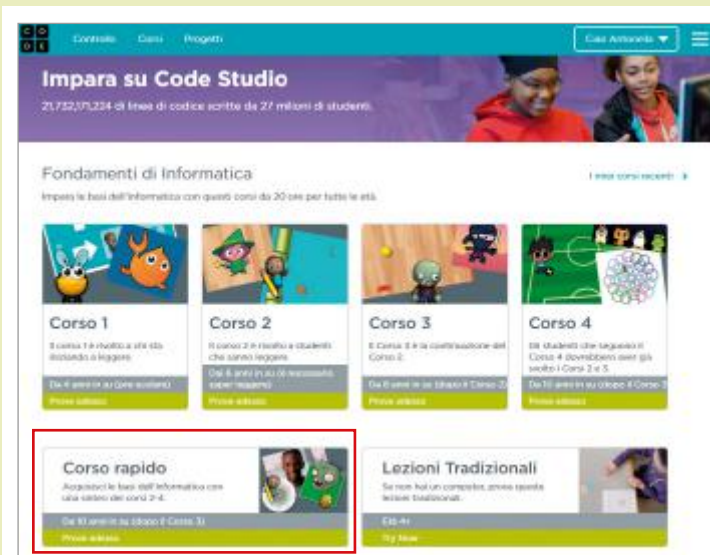


Figura M
Alcuni dei corsi disponibili in Code.org



Figura N
Esempio di attestato Corso rapido

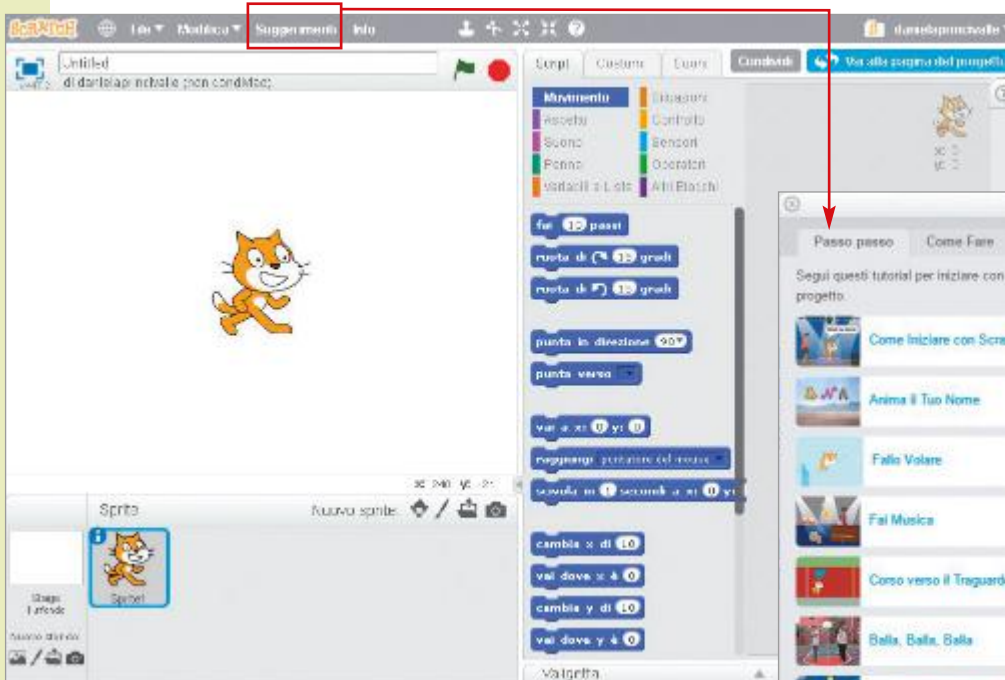


Figura O
Ambiente Scratch



Figura P
Accesso all'ambiente

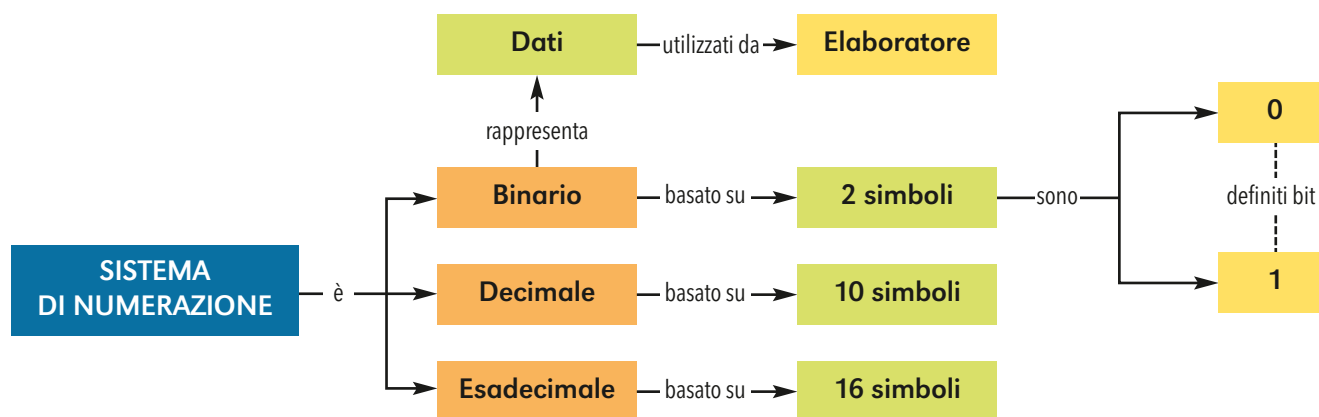


Figura Q
Esempio di tutorial

Concluso il percorso **L'Or del Codice**, dal menu **Corsi** del sito **Code.org** ti suggeriamo di svolgere il **Corso rapido** (Figura M), il cui tempo di esecuzione totale è stimato in 20 ore, alla fine del quale ottieni un ulteriore attestato (Figura N). Un linguaggio di programmazione visuale che offre un ambiente che ricorda quello proposto nei percorsi **Code.org** è **Scratch** (Figura O). Accedendo al sito <https://scratch.mit.edu/> e facendo clic su **Provalo** (Figura P) e poi, nel menu, su **Suggerimenti** (Figura O), si ottiene un elenco di tutorial che, mediante animazioni, guidano passo passo all'uso (Figura Q).

I sistemi di numerazione decimale, binario, esadecimale

La gestione dei dati da parte di un computer avviene tramite la loro codifica in **formato digitale binario**, cioè in sequenze di **0** e **1**. Per capire come ciò avvenga, partiamo dall'esame dei **sistemi di numerazione**, in particolare di 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.

$$\begin{aligned} 584 &= 5 * 10^2 + 8 * 10^1 + 4 * 10^0 \\ 845 &= 8 * 10^2 + 4 * 10^1 + 5 * 10^0 \end{aligned}$$

→ 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

Per accendere la luce in una stanza si preme l'interruttore che fa accendere la lampadina. In tal caso si hanno due soli possibili stati: a circuito elettrico chiuso la lampadina è accesa, a circuito elettrico aperto la lampadina è spenta.

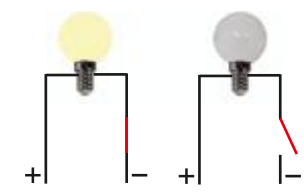


Figura 5
Esempio di circuito chiuso e aperto

L'esempio 6 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.

TEST 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 binario 1101001 in decimale.

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

CURIOSITÀ

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

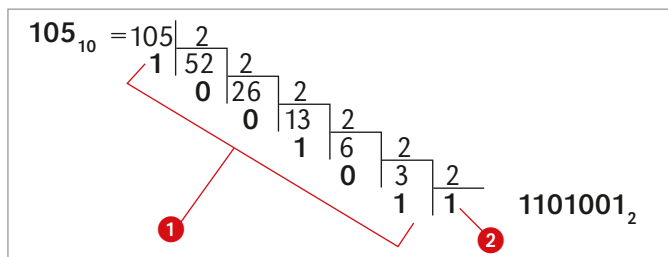
Convertire un numero decimale in binario

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. ←

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

Nella tabella a lato rapportiamo i simboli del sistema esadecimale al loro valore in decimale.

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.

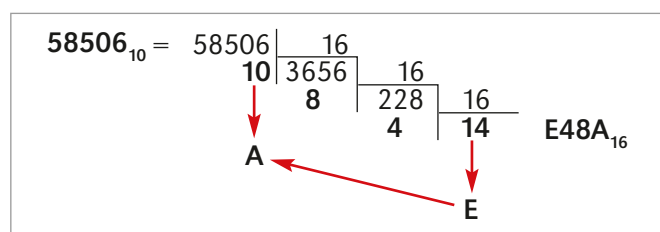
$$\begin{aligned} D3F_{16} &= D * 16^2 + 3 * 16^1 + F * 16^0 \\ &= 13 * 256 + 3 * 16 + 15 * 1 \\ &= 3328 + 48 + 15 \\ &= 3391_{10} \end{aligned}$$

TEST AL VOLO

Il numero esadecimale A1 corrisponde al numero decimale:
 161
 162

ESEMPIO 10

Convertiamo il numero decimale 58506 in esadecimale.



TEST 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.

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.

CURIOSITÀ

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

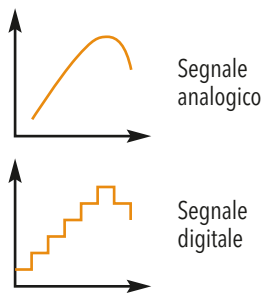


Figura 6
Rappresentazione grafica di segnale analogico e digitale

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.

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**”).

CURIOSITÀ

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.

CURIOSITÀ

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.

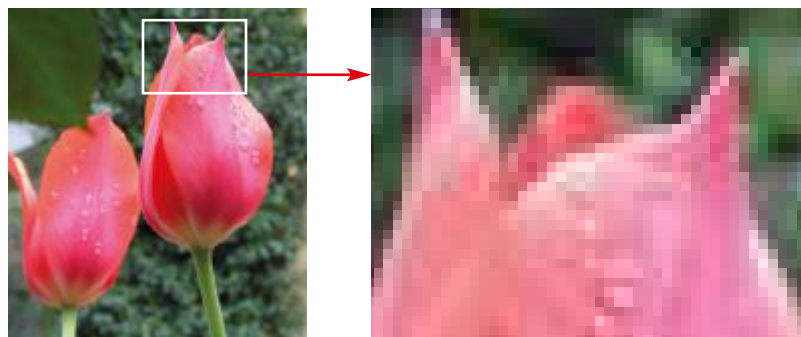
A lato sono mostrati alcuni formati associati al tipo di compressione applicata a file di immagini. La grande diffusione di dispositivi di elaborazione che gestiscono informazioni in formato digitale ha fortemente favorito la traduzione in termini digitali di grandezze e fenomeni analogici.



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 delle stesse. Se ingrandite, risultano sfuocate ed è riconoscibile la quadratura (*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





Rifletto...

Osserva le pagine da 2 a 16, 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...

Prova a strutturare un testo in forma grafica e testuale seguendo le istruzioni sotto riportate.

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:

- cerchia 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
