didattica delle scienze 115

pubblicazione bimestrale
dell'Editrice La Scuola
25186 Brescia

gennaio
1985
anno XX

didattica delle scienze 115

Bimestrale per l'insegnamento delle scienze e della matematica

Direttore Mauro Laeng, docente di Pedagogia all'Università di Roma
Redattore Giuseppe Luciano

gennaio 1985

Sommario

3 CARLO ANTONINO PRESTIPINO, Termodinamica e filosofia. Parte quarta

7 BRUNO DE SIMONE - GIOVANNA ESPOSITO DI LORENZO, Osservazioni geologiche sulle isole Galápagos

10 WALTER BODDI - DANIELA CATARZI - FABIO OLMI - GIOVANNI PEZZATINI, Insegnamento delle scienze ed educazione alla salute 2. Il rapporto medicina-salute. Seconda parte

35 GIULIA LEVI, Lavoisier e « l'esprit d'exactitude ». Vita, ricerche e metodo di un grande chimico

40 ANGELA FIORINI RAMPA, Vecchi strumenti per nuovi problemi. Processi cognitivi e apprendimento della matematica a livello di scuola elementare

44 Notiziario

46 Recensioni

Fascicolo di 28 pagine più inserto redazionale

Ad ogni comunicazione o richiesta riguardante la rivista i sigg. abbonati sono pregati di allegare una copia del talloncino-indirizzo col quale la rivista stessa viene loro spedita.

Inserto

La seconda parte dell'inserto *L'informatica nella scuola d'oggi* si articola in tre punti. Nel primo vengono descritte le caratteristiche e l'evoluzione dei linguaggi di programmazione; nel secondo viene trattato il software nelle sue applicazioni più attuali; nel terzo sono presentati alcuni esempi di software didattico che potranno essere utili per un lavoro in classi provviste di microcomputer. L'inserto è come sempre completato da un utile Glossario che raccoglie i termini caratteristici del linguaggio di questa nuova scienza.

In copertina

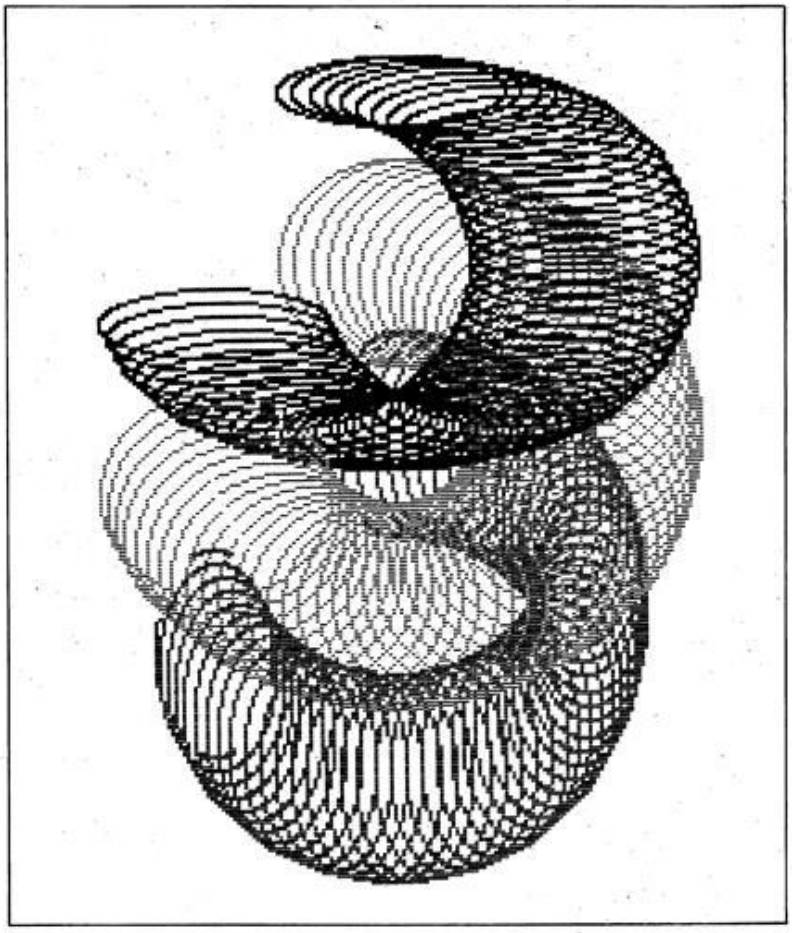
Tessitore di Baya maschio (*Ploceus philippinus*). Col nome di tessitori sono note alcune specie di Ploceidi che si distinguono in seno a questa famiglia per le loro maggiori dimensioni. Essi devono il loro nome alla capacità di tessere con molta arte i propri nidi. Una delle caratteristiche tipiche dei tessitori è il becco, che è molto grande e convesso e si prolunga fino sulla fronte. Il *Ploceus philippinus*, lungo circa 13 cm, ha un bellissimo piumaggio bruno scuro con striscette rosse sulle parti superiori, rossastre su quelle inferori del corpo, giallo oro e nero sul capo e sulla gola; esso rappresenta l'unica specie non africana del gruppo, ed è diffuso ampiamente in tutto il Sud-Est asiatico. Il tessitore di Baya costruisce stupendi nidi a forma di fiasco, dotati di una lunga galleria d'ingresso. I tessitori vivono in colonie numerose e nonostante debbano essere considerati uccelli dannosi alle colture, non sono oggetto di caccia e riscuotono da parte dell'uomo una certa simpatia, come accade in Europa per i passeri comuni. Sono apprezzati e ricercati ospiti delle voliere per i loro vivaci colori e il loro incessante, allegro cinguettio.

Pubblicazione bimestrale - Anno XX - n. 6 fascicoli all'anno - Direttore Responsabile: Giusto Marchese - Autorizzazione del Tribunale di Brescia n. 228 del 31 marzo 1965 - Spedizione in abbonamento postale - Gruppo IV/70 - **Direzione, Redazione, Amministrazione:** Editrice La Scuola - S.p.A. - 25186 Brescia, Via Luigi Cadorna, 11 - Conto corrente postale n. 11353257 Tel. centr. (030) 29 93.1 - Tel. Ufficio Abbonamenti (030) 29 93.286 - 29 93.246 - Telex 300836 SCUOLA.
Filiali: 40131 Bologna (Via L. Cipriani, 5, tel. (051) 521090 - telex 531141 SCUOBO); 20136 Milano (Viale Bligny, 7, tel. (02) 8370271 - telex 331836 SCUOMI); 00193 Roma (Via Crescenzio, 23, tel. (06) 655179 - 6543989 - telex 614259 SCUORO); 80137 Napoli (Via S. Elia ai Miracoli, 19/21, tel. (081) 441.200-441.934 telex 720399 SCUONA); 70124 Bari (Via Giulio Petroni, 21 A/E, tel. (080) 228647 - telex 810391 SCUOBA).
Abbonamento annuo 1984-85: L. 18.000 (estero L. 22.000). Un fascicolo L. 3.200 (arretrato il doppio).
Stampa: OFFICINE GRAFICHE LA SCUOLA - 25186 BRESCIA.

Giovanni Corsi - Giuseppe Bleiner

L'INFORMATICA NELLA SCUOLA D'OGGI

2. I LINGUAGGI DI PROGRAMMAZIONE. IL SOFTWARE. ALCUNI ESEMPI DI SOFTWARE DIDATTICO



Didattica delle Scienze 115 - Inserto redazionale monografico « L'informatica nella scuola d'oggi » n. 2.
Nell'illustrazione a lato: grafico-studio eseguito con un computer IBM.

1. Informatica e Cibernetica

Il termine « Informatica » fu coniato nel 1962 da Philippe Dreyfus, Direttore Generale del *Centre d'Analise et Programmation*, sezione dell'*Association Française de Calcul et Traitement de l'Information*. È una scienza che ha per oggetto lo studio dell'elaborazione automatica delle informazioni ottenute per mezzo dei computer. Etimologicamente il termine Informatica deriva dall'unione di due parole: " informazione " ed " automatica ", trattamento, quindi, dell'informazione mediante automi.

L'Informatica è collegata ad un'altra importante scienza, la Cibernetica (dal greco, arte del governare) che, secondo il dizionario Devoto-Oli, è « *un ramo sviluppatosi recentemente nella scienza pura e applicata, che si prefigge lo studio e la realizzazione di dispositivi e macchine capaci di riprodurre le funzioni del cervello umano per mezzo della trasformazione di segnali di comando e di controllo nei circuiti elettrici e nei sistemi meccanici* ». La Cibernetica è legata agli studi sull'intelligenza artificiale ed alla Robotica.

La parola Cibernetica fu introdotta nel 1948 dallo scienziato americano Norbert Wiener, che ha avuto il grande merito di aver capito che negli anni a venire la rivoluzione informatica avrebbe determinato una nuova mappa del potere, non più collegato al possesso del territorio o ai mezzi di produzione, ma al controllo dell'informazione.

È in questa teoria che intravvediamo l'importanza dell'Informatica e la necessità del suo dominio da parte dell'uomo, dominio che deve essere finalizzato ad un miglioramento della qualità della vita. Ma il controllo del potere dell'Informatica sarà possibile se la maggioranza degli uomini avrà scienza e conoscenza del fenomeno informatico. Ci sembra doveroso quindi aggiungere che solo la scuola, rinnovata nelle sue strutture e nella mentalità, potrà fornire i contenuti culturali e le metodologie per attuare questo controllo, generando una nuova cultura tecnologica di cui dovrà essere permeata la società del duemila.

2. Il linguaggio di programmazione come mezzo di interazione uomo-computer

L'interazione uomo-computer avviene attraverso i linguaggi di programmazione che ci consentono di comunicare alla macchina i procedimenti algoritmici con i quali è possibile impostare e risolvere un problema dato, in un numero finito di passi.
L'algoritmo, però, deve essere comunicato al computer mediante un linguaggio comprensibile sia dal computer che dall'uomo. I linguaggi di programmazione permettono questa interazione mediante i programmi.
Va chiarito altresì che l'algoritmo descrive lo sviluppo logico di un processo, indipendentemente dai linguaggi di programmazione e dal tipo di *hardware;* il programma, invece, va scritto in un determinato linguaggio.
Come i linguaggi naturali anche i linguaggi di programmazione hanno delle regole precise da rispettare. Perciò per poter esprimere un contenuto è necessario un " alfabeto " che è composto di simboli alfanumerici e di simboli grafici speciali. Esiste poi una " grammatica " con regole da rispettare, infine una " sintassi " che ci consente di riconoscere in quale linguaggio è stato scritto il programma.
A differenza del vocabolario di un linguaggio naturale che è ricchissimo di vocaboli (decine e decine di migliaia) con i quali possiamo creare e comunicare qualsiasi idea o sentimento, il vocabolario di un linguaggio di programmazione è molto limitato, appunto perché il computer non ha sentimenti o stati d'animo da rappresentare. Perciò una cinquantina di parole sono sufficienti per rappresentare qualsiasi contenuto da elaborare.
I linguaggi di programmazione si dividono in due grandi categorie: linguaggi non evoluti *(low level languages)* e linguaggi evoluti *(high level languages):* i primi, orientati alla macchina, hanno una struttura ed una logica distante da quella umana e sono caratterizzati da un modo di comunicare molto prossimo al linguaggio macchina; i secondi, invece, orientati al problema, sono più vicini al linguaggio dell'operatore e in genere hanno caratteristiche tali da

renderli specifici più o meno al problema in oggetto. Tra i linguaggi non evoluti ascriviamo il *Linguaggio Macchina* e l'*Assembler*, tra quelli evoluti tutti gli altri linguaggi che si sono affermati negli ultimi venti anni: il *Basic*, il *Logo*, l'*Algol*, l'*Ada*, il *Lisp*... Nell'ambito dei linguaggi è da sottolineare il problema dei " dialetti ", la diversità cioè delle istruzioni pur nell'ambito dello stesso linguaggio. Ad esempio, un programma scritto nel *Basic Apple* non gira sul Commodore, che pure parla il *Basic*, perché il *Basic Apple* è leggermente diverso dal *Basic Commodore*.

Prima del 1950 l'unico linguaggio comprensibile dai computer era il *Linguaggio Macchina* (LM), dipendente dalle caratteristiche *hardware* del sistema, espresso secondo un codice binario, ottale o esadecimale.

Questo tipo di programmazione era difficile e complesso e conosciuto solo da pochi addetti ai lavori. Poiché non esistevano le tastiere, le istruzioni contenute nei programmi venivano immesse una alla volta mediante interruttori che nelle diverse condizioni di accesso (1) o spento (0) rappresentavano la codifica binaria delle istruzioni. Si sentì allora l'esigenza di individuare un linguaggio più semplice e più vicino alla logica dell'utilizzatore. Nacquero, così, i linguaggi simbolici che hanno bisogno di un interprete che legga le istruzioni, verifichi la loro correttezza e, se tutto è a posto, traduca automaticamente le istruzioni del programma in LM, dando il via al programma.

Il primo di questi linguaggi è stato l'*Assembler*, un linguaggio non evoluto, classificato come livello 1. La programmazione in *Assembler* presuppone una profonda conoscenza dell'*hardware* e del *software* del computer. È conosciuto solo dai più esperti programmatori e viene utilizzato quando si ha l'esigenza di sfruttare appieno le caratteristiche dell'*hardware* o per velocizzare determinate procedure o quando si presenta la necessità di collegare dispositivi di I/O non standard.

Successivamente si cominciarono a studiare linguaggi più evoluti e più vicini al linguaggio naturale, come il *Fortran*, l'*Algol*, il *Cobol*, il *Basic*... Questi linguaggi di alto livello vengono convertiti in codice macchina mediante determinati programmi denominati *interpreti* o *compilatori;* abbiamo così i programmi interpretati ed i programmi compilati.

L'*interprete*, normalmente implementato nella memoria ROM del computer, traduce automaticamente passo passo in LM il programma scritto in un linguaggio ad alto livello. In seguito al " *run* ", il programma viene tradotto ed eseguito. L'*interprete Basic*, presente in quasi tutti i microcomputer, è un esempio di quanto detto. Il *compilatore*, invece, esegue la traduzione di tutto il programma prima della sua esecuzione.

Le due tecniche evidentemente hanno vantaggi e svantaggi: i programmi interpretati consentono di osservare il risultato istruzione per istruzione, ma sono lenti; i programmi compilati non possono essere seguiti passo passo, ma sono velocissimi, almeno cinquanta volte più di un programma interpretato, perché nel momento del " *run* " il programma è stato già tutto tradotto.

Esistono programmi compilatori per molti linguaggi (ad es., l'*Astro compiler* e il *Pet speed* per il CBM 64), mentre alcuni linguaggi, come il *Cobol* o il *Fortran*, sono nati proprio come compilatori. Un altro non indifferente vantaggio dei compilatori è che, poiché tutte le istruzioni vengono codificate in LM, è quasi impossibile risalire al programma " sorgente " e questo è un valido mezzo di protezione dei programmi stessi dai " pirati " del *software*.

3. Linguaggi di programmazione implementati sui personal più diffusi

Quando il costruttore dell'*hardware* dichiara che su di un computer sono disponibili determinati linguaggi di programmazione, significa che nel *software* di base del computer è presente un interprete capace di tradurre automaticamente in LM quel determinato linguaggio.

Attualmente i microcomputer più diffusi parlano almeno il *Basic;* quelli più sofisticati, mediante schede aggiuntive o da *software*, sono in grado di parlare anche linguaggi molto potenti e di livello elevato come il *Lisp*, il *Logo*, il *Fortran* o il *Cobol*, il *Pascal*. Per il Commodore 64, ad esempio, sono disponibili, oltre al *Basic standard Commodore*, il *Comal*, il *Logo*, il *Pilot*, l'*UCDS Pascal*, il *Forth*, l'*Assembler*...; altrettanto vale per l'Apple II.

Va aggiunto, anche, che connettendo una scheda o un apposito *cartridge*, contenenti un secondo processore, lo Z 80, i due microcomputer appena citati acquisiscono la possibilità di usare l'enorme *libreria software* scritta con *Sistema Operativo* CPM ed altri codici Z 80, espandendo notevolmente la quantità di *software* disponibile. Al microcomputer Acorn BBC

Il Commodore, uno dei personal più diffusi.

invece, che utilizza il microprocessore Z 80 di 64 K di memoria RAM, si può connettere un secondo processore, il 6502 implementato nell'Apple II, nell'Atari e nel Vic 20.

4. Caratteristiche di alcuni linguaggi di programmazione

ADA: linguaggio di programmazione ad alto livello dell'ultima generazione, fu creato in Francia da una équipe di studiosi della Honeywell, diretta da J. Ichbiah. Fu commissionato dal Ministero della Difesa degli Stati Uniti allo scopo di creare un sistema informativo omogeneo per controllare e coordinare, con un solo linguaggio di programmazione, tutte le attività dell'esercito, della marina e dell'aviazione.
L'*Ada*, mediante la tecnica delle procedure, ha la possibilità di collegare i sottoprogrammi di cui è composto il programma principale in modo semplice, ma efficace.

ALGOL (ALGorithmic Oriented Language): linguaggio destinato ad essere applicato agli algoritmi, pensato come universale (in origine, infatti, fu chiamato *IAL, International Algebraic Language*) indipendente cioè dal tipo di macchina, scaturì dal lavoro di una commissione mista americano-tedesca negli anni intorno al 1960.
Algol è un linguaggio eminentemente matematico e scientifico, strutturato a blocchi, adatto a descrivere processi di calcolo.
Nel 1968 ne uscì una nuova versione, l'*ALGOL 68*, più potente, versatile e semplice del linguaggio originario, tuttavia la potenza del *Pascal* lo ha in parte soppiantato.

APL (A Programming Language): creato da Kennet E. Iverson, è finalizzato ad applicazioni scientifiche ed alle simulazioni con modelli, in particolare per il trattamento dei vettori e delle matrici. È molto potente, ma complesso da utilizzare. Ha numerosi seguaci in tutto il mondo, ma a causa della sua complessità e per il fatto di richiedere un *hardware* particolare, pensiamo che non diventerà mai un linguaggio popolare.

ASSEMBLER: linguaggio simbolico di programmazione di basso livello, è indirizzato alle caratteristiche *hardware* della macchina. È molto più veloce del *Basic* ed è capace di agire direttamente sull'insieme delle *routines* presenti nel microprocessore. È difficile da utilizzare e perciò conosciuto solo dai più esperti programmatori.
A differenza del *Basic* in cui a ciascuna istruzione corrispondono una serie di istruzioni in LM, nell'*Assembler* ad ogni istruzione ne corrisponde un'altra in LM e viceversa. Esistono apposite tabelle per effettuare la conversione. Per utilizzare questo linguaggio è necessario un programma *Basic* detto *Assemblatore* che traduce le istruzioni *Assembler* in codice macchina.

BASIC (Beginner's All-Purpose Symbolic Instruction Code): linguaggio interpretato, derivato dal *Fortran*, è stato creato intorno al 1964 dalla General Electric, in collaborazione col Dartmouth College (principali artefici furono Thomas E. Kurtz e G. Kemeny), per essere destinato ai principianti e per l'insegnamento della programmazione. È un linguaggio semplice, colloquiale, ma potente. Col tempo è stato utilizzato per applicazioni di ogni tipo tanto da essere diventato il linguaggio più diffuso sia sui "*mainframe*" che sui microcomputer.
Il *Basic* è un linguaggio ottimale per la realizzazione di programmi brevi; mostra i suoi limiti, però, quando il programma supera il migliaio di istruzioni. Inoltre, poiché il *Basic* è un linguaggio "interpretato" (cioè una istruzione, ogni volta che viene eseguita, deve essere tradotta in LM, a differenza dei linguaggi "compilati" dove la traduzione in LM avviene una volta per tutte) gira con lentezza. Essendo, anche, un linguaggio lineare, quando il programma è lungo e complesso è difficile da comprendere. Ha il vantaggio, però, di consentire la scrittura del programma passo passo e di fornire i risultati parziali. A causa della sua interattività è molto utilizzato nella didattica.
Secondo alcuni la macchinosità della programmazione che può determinare cattive abitudini nei principianti e la presenza di numerosi dialetti che impediscono di fatto lo sviluppo di un linguaggio standard, hanno determinato un atteggiamento un po' ostile verso il *Basic*, ciò tuttavia non limita di fatto il successo di questo linguaggio.

LINGUAGGIO C.: è stato elaborato presso i Bell Laboratories da Dennis Ritchie intorno al 1972 per essere utilizzato nei *Sistemi Operativi* (SO) *Unix* della Digital Equipment Corporation. È caratterizzato dalla possibilità di operare a basso livello pur potendo essere ascritto fra i linguaggi di alto livello. Caratteristica principale del *Linguaggio C.* è la semplicità di scrittura del programma e la possibilità di elaborare un codice macchina compatto e potente. Per alcuni versi il *linguaggio C.* presenta delle somiglianze col *Pascal* che tuttavia non consente accessi a *routines* a basso livello, a differenza primo.

COBOL (COmmercial Business Oriented Language): linguaggio orientato ad applicazioni commerciali. Fu creato da un gruppo di programmatori dell'università di Pennsylvania intorno agli anni sessanta. È uno dei linguaggi preferiti per impieghi gestionali e commerciali con grandi quantitativi di dati. Ha bisogno di memorie di massa di grandi capacità. È semplice da usare ed è dotato di una struttura rigida. Le sue istruzioni sono verbali più che matematiche. È un linguaggio standard che consente ai programmi scritti in *Cobol* di girare anche su macchine strutturate con *hardware* diverso.

COMAL: molto diffuso in Svezia dove è stato creato, è utilizzato soprattutto nelle scuole di calcolo. Deriva dal *Basic* e dal *Pascal* da cui ha ereditato le migliori opzioni.

FORTH: complesso linguaggio di programmazione, considerato della quarta generazione, è stato ideato da Charles Moore intorno al 1970. Permette la definizione di comandi personalizzati. Somiglia al *Logo*

perché è un linguaggio funzionale e interattivo. Utilizza la *Notazione Polacca Inversa*, molto usata nelle piccole calcolatrici tascabili. È caratterizzato da elevata velocità di elaborazione.
Prima di essere implementato sui microcomputer odierni, fu utilizzato per applicazioni sui grandi sistemi e dagli astronomi per la sua elevata capacità di simulazione dei fenomeni in tempo reale. È più vicino al linguaggio *Assembler* rispetto agli altri linguaggi di alto livello.

FORTRAN (FORmula TRANslator): linguaggio di programmazione compilato di alto livello, ideato in ambiente IBM da J. Backus intorno al 1957. Il suo uso è eminentemente matematico, scientifico e tecnologico. Ha conosciuto il periodo di maggiore splendore negli anni '60. Ha bisogno di una memoria centrale molto capiente; è dotato di ricorsività e permette la stesura di sottoprogrammi da impiegare nel programma stesso. Essendo poco interattivo, è stato soppiantato dal *Basic*.

LISP (LISt Processor): linguaggio elaboratore di liste, è uno tra i più antichi linguaggi di programmazione.
È stato creato da John Mc Carthy nel 1960 presso il laboratorio A.I.P. del MIT come linguaggio scientifico finalizzato alla manipolazione di formule matematiche. È caratterizzato da una alta componente di simbolicità, genera facilmente dati strutturati ad albero ed è adatto anche ad applicazioni educative. È utilizzato soprattutto nella robotica e nelle simulazioni relative all'intelligenza artificiale.
Dal *Lisp* sono derivati altri due linguaggi che hanno la stessa capacità: l'*Apl* e lo *Snobol*.

LOGO (dal greco: discorso): i primi studi sul *Logo* furono finanziati dalla National Science Foundation. Successivamente questo linguaggio, semplice ma potente, fu elaborato da Seymour Papert, dell'Artificial Intelligence Laboratory del MIT (Massachussets Institute of Technology), a Cambridge nel Massachussets, mentre conduceva studi sull'intelligenza artificiale. È indirizzato soprattutto ai bambini e si ispira agli studi di Piaget.
La filosofia del *Logo* si basa sul principio che è lo stesso bambino a programmare la macchina e non la macchina a programmare il bambino (secondo Papert i bambini possono imparare a programmare in *Logo* così come imparano a camminare o a parlare).
Per Giovanni Lariccia il *Logo* rappresenta nell'educazione una possibile rivoluzione culturale, maturata all'interno di una linea di profonda tradizione scientifica; è parte, quindi, di una nuova filosofia educativa.
Il *Logo* è anche un ambiente di apprendimento in cui il bambino, attraverso procedure dette " primitive ", può definire proprie procedure da utilizzare nella costruzione di immagini di fantasia, dominando la macchina.
Senza condizionamenti di sorta e con semplici comandi il computer può essere interfacciato con una sorta di semplice automa, detto *Turtle*, che somiglia ad una tartaruga stilizzata e che consente di vedere in tempo reale gli spostamenti dell'oggetto.

Di questo linguaggio, ormai diffuso in tutto il mondo, esistono diverse versioni che girano sui principali computer in commercio. La versione italiana è stata curata da Giovanni Lariccia nelle versioni per i microcomputer Texas TI 99 4/A, Apple II e Commodore 64.

PASCAL: derivato dal *Fortran* e dall'*Algol*, fu ideato da Niklaus Wirth, docente al Politecnico di Zurigo, intorno al 1970 come linguaggio didattico per programmatori. La versione che ha avuto più successo è l'*UCDS* (*University of California, San Diego*). Come diffusione viene appena dopo il *Basic*.
La sua architettura deriva dall'*Algol*, ma è più potente e più semplice da utilizzare, sebbene non contenga alcune opzioni presenti nell'*Algol*.
Il *Pascal* consente la programmazione strutturata (tecnica " *top down* ", modularità...) e quindi la trattazione di argomenti molto complessi, ma facilmente leggibili e comprensibili. Consente, anche, la creazione di istruzioni definite dall'utente e buona è la tecnica per effettuare le operazioni di I/O. Il listato del programma si presenta in modo raffinato ed elegante. È il più utilizzato nelle università.

PEARL: linguaggio di alto livello, in uso nella Germania Federale, impiegato per controlli di processo in tempo reale.

PL/I (Programming Language N. 1): è un linguaggio compilativo, modulare, molto potente e complesso, non dipendente dalle caratteristiche *hardware* che, secondo gli autori, doveva essere la sintesi dei linguaggi *Algol*, *Cobol* e *Fortran* con l'ambizione di sostituirli.
È adatto ai più svariati campi di applicazione, numerici e non numerici, in particolare nel campo commerciale e scientifico; consente la gestione di processi paralleli, cioè la esecuzione simultanea di diversi insiemi di istruzioni.
Non è molto diffuso perché, non potendo essere implementato sui piccoli sistemi, non ha trovato il favore della maggioranza dei costruttori di *hardware*, a parte l'IBM che ne è il principale sostenitore.

La tartaruga (*Turtle*) sistema computerizzato di ausilio didattico.

PILOT (Programmed Inquiry Learning or Theaching): realizzato intorno al 1968 da una équipe di studiosi dell'Università di California, sotto la direzione di John A. Starweather, è un linguaggio autore che consente ai docenti, che hanno scarse cognizioni d'informatica, di realizzare in modo autonomo unità didattiche in modo semplice ed efficace. Il *Pilot* è dotato di grafica, suoni e colori. Un semplice repertorio d'istruzioni consente di operare con sicurezza e potenza.

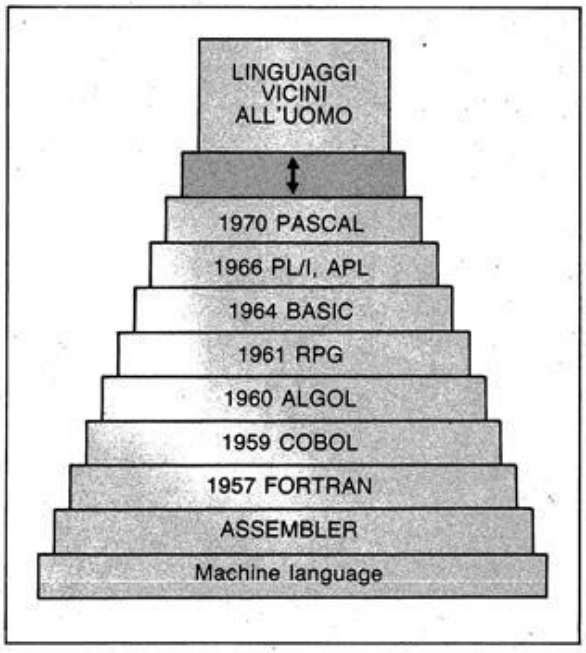
RPG (Report Program Generator): concepito in ambiente IBM negli anni sessanta è finalizzato al trattamento ed alla generazione di rapporti e di tabelle di dati standard. È molto utilizzato nel campo gestionale.

SNOBOL (StriNg Oriented SymBOLyc Language): linguaggio simbolico orientato alle stringhe, fu sviluppato nei laboratori Bell all'inizio degli anni sessanta ed è finalizzato alla manipolazione di testi. È applicato particolarmente alla compilazione, all'analisi ed alla traduzione di linguaggi di programmazione e nel calcolo combinatorio.
Molto utilizzato nei programmi CAI *(Computer Aided Instruction)* e nella teoria dei giochi, consente una facile gestione dei suoni e la simulazione di automi.

5. Il Software

Il *software* è l'insieme dei programmi che permettono all'*hardware* di funzionare per realizzare le funzioni per cui è stato costruito. Esso è composto da una serie di istruzioni che, secondo un rigido ordine sistematico, provvede ad accendere o a spegnere (0/1 - sistema di codifica binaria) le migliaia di microinterruttori contenuti nel *chip*. Per ottenere questo è necessario che le istruzioni siano date in modo comprensibile al computer, occorre cioè codificare le istruzioni in LM. Tutto il *software* prodotto in linguaggi evoluti compie questa traduzione automaticamente.
Il programma per poter " girare " deve essere introdotto nella memoria centrale del computer e questa operazione può essere fatta con la digitazione diretta del programma o caricandolo da una memoria magnetica di massa (*floppy disk*, cassetta). Il programma può essere, anche, introdotto in modo permanente nella memoria ROM del computer ed in tal caso l'accesso è istantaneo. Il caricamento da *floppy disk* richiede un certo tempo, seppur breve; il caricamento da cassetta, invece, richiede un tempo molto più lungo.

Schema della successione di alcuni dei principali tipi di linguaggi di programmazione.



6. I sistemi operativi

Si definisce *Sistema Operativo* (SO) il *software* di base che consente la facile gestione del sistema (controllo del computer e delle periferiche: memorizzazione delle informazioni sul *floppy;* stampa delle elaborazioni; gestione delle operazioni di I/O ...). Il SO costituisce l'interfaccia con l'utilizzatore. Normalmente esso è implementato nella memoria ROM del computer ed è inaccessibile all'operatore.
Nei microcomputer dotati di unità a disco, parte del SO viene utilizzato per gestire appunto il funzionamento di questa periferica: viene chiamato *DOS (Disk Operating System,* cioè *Sistema Operativo per Disco).*
Il DOS può essere implementato nella memoria ROM del computer, come nello Spectrum; nella ROM dei *disk driver* intelligenti come quello del Commodore che contiene un proprio microprocessore; nella memoria RAM del computer, quando essi sono dotati di molta memoria e quando si ha l'esigenza di utilizzare diversi sistemi operativi.
Alcuni SO come il CP/M *(Control Program for Microprocessor),* lo MS-DOS, l'UNIX, lo UCSD... sono caratterizzati da differenti possibilità. Ad esempio, nelle ultime versioni, alcuni di questi SO consentono di effettuare il *multitasking,* cioè la gestione multipla contemporanea della macchina.
Per quanto riguarda l'origine dei sistemi operativi ricordiamo che il CP/M è stato inventato da Gary Killdal, un hobbysta californiano che, visto il successo del sistema, fondò la Digital Research, che tuttora si occupa della diffusione di questo sistema e delle versioni successive.
Il noto sistema MS-DOS della Microsoft fu acquistato da un'altra ditta. Le versioni successive, tuttavia, sono tutte della Microsoft.
L'UNIX è un sistema realizzato dalla Bell Laboratories ad opera di K. Thomson ed è caratterizzato da una notevole duttilità.
L'APPLE-DOS è un sistema implementato sulle macchine della società di Cupertino ed è caratterizzato da semplicità e facilità d'uso, ma non è un vero sistema operativo.

7. Programmi di simulazione

I programmi di simulazione hanno lo scopo di rappresentare simbolicamente e nel modo più fedele possibile la realtà. Sono molto utilizzati nella didattica, soprattutto delle materie scientifiche. È evidente il vantaggio di poter simulare esperienze con materiali pericolosi o esplosivi o molto costosi. Interessante è anche la simulazione di macchine complesse inaccessibili, di processi industriali quali quelli che si verificano in una acciaieria o in una industria petrolchimica sino a giungere alle reazioni termonucleari.

Può essere simulato e studiato anche il funzionamento di un organo interno al nostro corpo o il viaggio nello spazio di una navicella. Molto diffusi sono i simulatori di volo *(flight simulator)* che danno l'impressione di guidare veramente un aereo e quelli di eventi meteorologici o di battaglie storiche.

8. Programmi applicativi più diffusi ed utili

8.1. *Spreadsheet*

Lo *spreadsheet* o lavagna elettronica, finalizzato al calcolo numerico, è uno dei programmi base (assieme ai *data base*, al *word processor* ed alla contabilità generale) che da soli giustificano l'uso e l'acquisto di un personal computer.

Lo *spreadsheet* si avvale di un sistema che suddivide il *display* in righe e in colonne. L'intersezione di un rigo con una colonna è chiamato " campo " ed è definibile mediante due coordinate (ad es. A4, B5). Ad ogni " campo " può essere assegnato un numero, una formula matematica o un nome.

L'utilità pratica di questo programma veramente notevole è che cambiando i valori di un dato, tutti gli altri, e talvolta si tratta di migliaia e migliaia di informazioni, vengono ricalcolati istantaneamente. Si pensi, perciò, ai vantaggi che ne derivano ad un manager (che prima faceva tutti questi calcoli con carta e penna) che deve aver sempre presente la situazione aziendale; si pensi alla programmazione della produzione industriale o a quella delle vendite. Né va sottovalutato l'utilizzo dello *spreadsheet* nella ricerca scientifica e nella simulazione di modelli computerizzati per la capacità che esso ha di rispondere alla domanda " cosa accadrebbe se ... ".

Gli *spreadsheet* più sofisticati, come il CALC RESULT, implementato sul Commodore 64, hanno la possibilità di visualizzare sul *display* il risultato delle elaborazioni in forma di grafici (istogrammi, diagrammi areali...) e di stamparli su carta oltre che memorizzarli su supporto magnetico. Molto buono è anche il MULTIPLAN della Microsoft di cui esistono versioni sia per l'Apple II che per il Commodore 64.

8.2. *Word processor*

Il *word processor* trasforma il computer in un piccolo ed autosufficiente centro stampa. Esso consente di creare e di manipolare con facilità ed a proprio piacimento qualsiasi testo: lettere, rapporti, dispense, libri. I testi possono essere memorizzati su supporto magnetico, conservati in archivio ed essere riutilizzati all'occorrenza.

Prima della stampa il testo può essere visualizzato sul *display* nello stesso modo in cui sarà stampato. Dopo questa operazione esso potrà essere stampato a nostro piacimento (tipo di carattere, numero di caratteri per linea, spaziatura, marginatura, giustificazione...) secondo l'impaginato progettato ottenendo un risultato prossimo a quello tipografico, se possediamo una stampante di qualità.

Molto noti sono l'EASY SCRIPT e il WORD CRAFT della Commodore e l'APPLE WRITER dell'Apple.

8.3. *Data base*

Il *data base* consente di gestire un archivio di dati e di ordinarli secondo criteri di nostra convenienza. Un tipico uso di questo programma è nella gestione di un indirizzario. Immessi i dati, in poco tempo possiamo ordinare le informazioni secondo l'ordine alfabetico (parziale o totale), ed ancora secondo la città, l'indirizzo, il cap, il numero telefonico... ottenendo infine la stampa di etichette autoadesive.

Lo stesso lavoro, prima dell'avvento del computer, veniva effettuato con schede ed a mano e richiedeva un tempo notevolissimo.

L'uso del *data base* richiede l'unità a dischi.

Un ottimo *data base* per microcomputer è il SUPERBASE della Commodore.

Immagine ripresa da un simulatore di volo. Il computer principale dei simulatori contiene le immagini delle piste dei principali aeroporti del mondo, che vengono proiettate su uno schermo. I comandi dei piloti passano attraverso il computer principale che modifica l'immagine di conseguenza.

GLOSSARIO

Albero: tipo di struttura gerarchica dei dati con al vertice la radice cui sono correlati tutti gli altri dati.

Assemblatore: programma incluso nel *software* di base che consente di tradurre il linguaggio *Assembler* in codice binario o LM.

Assembler: linguaggio simbolico di programmazione di livello 1, molto vicino al LM, consente di comunicare coll'*hardware* secondo una tabella di istruzioni più semplice del LM. V'è relazione 1 a 1 fra le istruzioni *Assembler* e quelle in LM.

Backup: operazione che consente la copia fedele di un supporto magnetico di memoria di massa.

Baud: misura la velocità di trasferimento dei dati in *bit per secondo.*

Benchmark: tipo di programma a test che consente di comparare le caratteristiche di due o più computer per valutarne l'efficienza.

Bootstrap (Bootstrap Loader): programma che permette di attivare la macchina abilitandone il SO.

Break: interruzione forzata di un programma.

Buffer: area riservata della memoria nella quale sono temporaneamente immessi i dati da trasferire successivamente in una periferica. Serve a compensare la diversa velocità di trasferimento dei dati nell'ambito del sistema. La presenza del *buffer* deriva dalla diversa velocità di elaborazione dei componenti il sistema e serve a renderne ottimale l'utilizzo. Ad esempio, mentre alcuni dati sono visualizzati attraverso la stampante, contemporaneamente possiamo riversarne nel *buffer* degli altri da elaborare successivamente.

Bug (letteralmente pulce): errore contenuto in un programma che gli impedisce di girare.

Campo (Field, Item): componente di un *record* contenente una unità d'informazione.

Catalogo: elenco dei *files* di una *base dati.*

Chiave (Key): carattere alfanumerico che serve alla identificazione di un *record.*

Compatibile: indica la possibilità di interfacciare componenti di un sistema con un sistema di marca diversa. Si riferisce anche al *software* che gira su macchine diverse, ma con lo stesso SO.

Compilatore o *interprete: software* che consente la traduzione automatica in LM di programmi realizzati in linguaggi simbolici o di alto livello.

CPM *(Control Program for Microprocessors):* è un Sistema Operativo. La sua larga diffusione consente alle macchine che lo utilizzano di poter disporre di una enorme libreria di *software.*

Crash (letteralmente crollo): fenomeno che indica il blocco del computer determinato da fattori ignoti.

Data base: è un programma che consente di gestire un archivio di dati e di ordinarli secondo criteri di nostra convenienza.

Debug: procedimento che consente di individuare gli errori nei programmi e di correggerli.

Device: dispositivo, apparecchiatura.

D.O.S. *(Disk Operating System): software* di base che controlla e consente il facile uso del *disk driver.*

Esadecimale: sistema numerico a base 16. Utilizza le cifre decimali da 0 a 9 ed i simboli alfabetici A, B, C, D, E, F.

File: sinonimo di archivio è costituito da un insieme di *record* con caratteristiche omogenee. È un insieme logico di informazioni, correlate fra loro, individuabili da un nome che permette di richiamarle in memoria al momento opportuno.

I/O *(Input/Output):* Ingresso/Uscita.

Linguaggio: mezzo che permette l'interazione uomo-computer. Mediante i linguaggi di programmazione possiamo comunicare al computer i procedimenti algoritmici che consentono di impostare e di risolvere un problema dato.

Loop: ciclo iterativo, tipo di algoritmo che ripete un certo numero di volte un ciclo o un processo.

Modello: rappresentazione simulata della realtà che si vuole studiare mediante una struttura a rete di variabili correlate tra loro.

Ottale: sistema numerico a base 8. Utilizza le cifre binarie da 0 a 7.

Package: è un insieme di programmi omogenei fra loro.

Pixel: ciascun punto che rende possibile la visualizzazione delle immagini su un *display.*

Random: sinonimo di casuale. Sta ad indicare la generazione automatica di numeri casuali o l'accesso diretto ai *file* memorizzati su di un *floppy disk.*

Record: blocco logico di informazioni che sono le più elementari che il computer può riconoscere. È costituito da un insieme di campi, unità logiche elementari. Più *record* costituiscono un *file.* Il *record* è individuabile mediante un numero od un nome.

Reset: pulsante che permette di interrompere un programma che sta girando e di cancellare la memoria del computer.

RS-232: interfaccia seriale standard.

Slot: dispositivo che consente di connettere al computer espansioni di memoria, interfacce...

Sort: algoritmo che permette di riordinare gli elementi di un archivio dati.

Spreadsheet: lavagna elettronica che consente di gestire grandi quantità di dati e di tabelle numeriche in modo semplice ed interattivo in cui tutti i dati sono collegati tra loro. Variando un dato automaticamente cambiano tutti gli altri. Consente di sapere " che cosa succederebbe se... ". È utilizzato nella pianificazione aziendale, nelle analisi di mercato, nelle proiezioni di ipotesi di lavoro, nella ricerca scientifica.

Stringa: insieme di caratteri considerati come unità.

Word Processor: programma che consente di elaborare testi, di correggerli facilmente e di memorizzarli su supporto magnetico. In pratica il *Word Processor* trasforma il computer in un piccolo centro stampa autonomo ed autosufficiente.

ALCUNI ESEMPI DI « SOFTWARE » DIDATTICO

LISTATO A

Autore: Paolo Coretti.
Nome del programma: Cellula e mitosi.
Linguaggio: Basic.
Hardware: Sinclair ZX SPECTRUM.
Scopo del programma: descrivere la struttura della cellula e il processo di riproduzione mediante mitosi.

```
  10 REM © P. CORETTI
  20 REM Cellula e Mitosi
 100 CLEAR 63999: GO SUB 1000: F
OR j=64000 TO 64119:: READ a
 110 POKE j,a: NEXT j
 120 DATA 42,101,92,237,91,99
 130 DATA 92,167,237,82,125,254
 140 DATA 10,218,139,40,205,148
 150 DATA 30,245,205,148,30,95
 160 DATA 241,87,33,1,1,66
 170 DATA 75,229,213,197,205,229
 180 DATA 34,193,209,225,205,84
 190 DATA 31,48,69,125,129,79,40
,19,229,213,197,205
 200 DATA 170,34,71,4,126,7
 210 DATA 16,253,230,1,193,209
 220 DATA 225,40,218,175,149,111
 230 DATA 254,1,32,210,124,126
 240 DATA 71,40,26,254,175,48
 250 DATA 22,46,1,75,229,213
 260 DATA 197,205,170,34,71,4
 270 DATA 126,7,16,253,230,1
 280 DATA 193,209,225,40,179,175
 290 DATA 148,103,254,1,32,171
 300 DATA 205,40,45,195,40,45
 310 RESTORE 1060: GO SUB 1060
 320 BORDER 0: INK 0: PAPER 0: C
LS
 330 BORDER 1:: PAPER 6: CLS
 340 GO SUB 1210
 350 CIRCLE 150,110,30: CIRCLE 1
40,100,10
 360 INK 4: RANDOMIZE 100+50*USR
 64000: RANDOMIZE 200+100*USR 64
000: INK 2: RANDOMIZE 140+100*US
R 64000
 370 PRINT AT 8,10; PAPER 4; INK
 1;"▲";AT 9,10;"▼";AT 15,5;"▲";A
T 16,5;"▼";AT 10,25;"▲";AT 11,25
;"▼"
 380 PRINT AT 5,5; PAPER 4; INK
 7;"╪";AT 12,13;"╪";AT 2,20;"╪";A
T 17,20;"╪"
 390 PRINT AT 5,5; PAPER 4; INK
 2;"⊂";AT 15,10;"⊂";AT 2,10;"⊂
";AT 15,25;"⊂"
 400 PRINT AT 2,25; PAPER 4; INK
 1;"*";AT 3,25;"*"
 410 PRINT AT 2,5; PAPER 4; INK
 3;".";AT 3,3;".";AT 7,6;".";AT 1
2,6;".";AT 16,15;".";AT 4,15;"."
;AT 15,27;"."
 420 BORDER 1: PAPER 6: INK 2
 430 PAUSE 0: GO SUB 1330: PRINT
 AT 9,20; PAPER 6; INK 0;"1": PR
INT AT 21,0; INK 0;"Nucleo": PAU
SE 0: PRINT AT 9,20; INK 6;"■":
GO SUB 1330
 440 PRINT AT 21,0; PAPER 6;"
                          ": PRI
NT AT 9,17; PAPER 2; INK 7;"2":
PRINT AT 21,0; INK 0;"Nucleolo":
 PAUSE 0: PRINT AT 9,17; INK 2;"
■": GO SUB 1330
 450 PRINT AT 21,0; PAPER 6;"
                          ": P
RINT AT 1,24; PAPER 4; INK 0;"3"
: PRINT AT 21,0; INK 0;"Centriol
i": PAUSE 0: PRINT AT 1,24; INK
4;"■": GO SUB 1330
 460 PRINT AT 21,0; PAPER 6;"
                          ": PR
INT AT 7,5; PAPER 4; INK 0;"4":
PRINT AT 21,0; INK 0;"Ribosomi":
 PAUSE 0: PRINT AT 7,5; INK 4;"■
": GO SUB 1330
 470 PRINT AT 21,0; PAPER 6;"
                          ": PRINT AT 2,
2; INK 0;"5";AT 21,0; PAPER 6; I
NK 0;"Membrana cellulare": PAUSE
 0: PRINT AT 2,2; INK 6;"■": GO
SUB 1330
 480 PRINT AT 21,0; INK 6;"
                          ";AT 7,10;
 PAPER 4; INK 0;"6";AT 21,0; PAP
ER 6;"Mitocondri": PAUSE 0: PRIN
T AT 7,10; INK 4;"■": GO SUB 133
0
 490 PRINT AT 21,0; PAPER 6;"
                          ";AT 11,13; PAP
ER 4; INK 0;"7": PRINT AT 21,0;
PAPER 6; INK 0;"Apparato del Gol
gi": PAUSE 0: PRINT AT 11,13; I
NK 4;"■": GO SUB 1330
 500 PRINT AT 21,0; INK 6;"
                          ";AT 14,10; PAPER
 4; INK 0;"8";AT 21,0; PAPER 6; I
NK 0;"Ergastoplasma": PAUSE 0: P
RINT AT 14,10; INK 4;"■";AT 21,0
; INK 6;"                    ": GO
SUB 1330: PAUSE 2: GO SUB 1330
 510 PAUSE 200
 520 PAPER 4: INK 2: CLS : GO SU
B 1210: INK 6: RANDOMIZE 15+87*U
SR 64000: RANDOMIZE 245+87*USR 6
4000
 530 REM PROFASE: GO SUB 9030
 540 GO SUB 1270: GO SUB 1330: P
RINT #0; INK 9;AT 0,0;"PROFASE:
cromosomi non visibili.": PAUSE
100
 550 PRINT #0;AT 0,0; INK 9;"I c
entrioli si portano ai poli. "
 560 BEEP .12,3: PRINT PAPER 4;
INK 4;AT 6,15;"■"; PAPER 4; INK
4;AT 6,17;"■": BEEP .5,3: PRINT
AT 8,11; PAPER 4; INK 6;"*";AT 8
,20; PAPER 4; INK 6;"*": BEEP .5
,3: PRINT INK 4; PAPER 4;AT 8,11
;"■";AT 8,20;"■": BEEP .5,3: PRI
NT PAPER 4; INK 6;AT 10,7;"*";AT
 10,23;"*"
 570 REM profase-metafase
 580 PAUSE 100
 590 PRINT #0;AT 0,0; INK 9;"MET
AFASE:Appaiono i cromosomi. "
 600 OVER 1: CIRCLE 110,93,5: OV
ER 0: PRINT AT 9,15; INK 4;"▄▄▄
";AT 11,13;"▄▄▄▄";AT 13,14;;"▄▄▄
▄"
 610 GO SUB 1300: GO SUB 1330: P
AUSE 100
 620 OVER 1: CIRCLE 128,87,30: C
IRCLE 128,87,32: OVER 0
 630 INK 2: PLOT 60,92: DRAW 125
,0
 640 PRINT OVER 1;AT 8,11; INK 4
;"■": PLOT OVER 1;60,92: DRAW 12
5,0,PI/3.5: PLOT OVER 1;60,92: D
RAW 125,0,-PI/3.5
 650 GO SUB 1300: PAUSE 20
 660 PRINT OVER 1; INK 4;AT 8,20
;"■";AT 13,14;"■";AT 13,17;"■":
PLOT OVER 1;60,92: DRAW 125,0,PI
/2: PLOT OVER 1;60,92: DRAW 125,
0,-PI/2: GO SUB 1300
 670 GO SUB 1330: PRINT #0;AT 0,
0; INK 9;"I cromosomi si portano
all'equa-tore della cellula.
                          ": PAUSE 200
680 REM anafase
690 PRINT #0;AT 0,0; INK 9;"ANA
```

(segue a pag. 24)

(segue da pag. 23)

```
FASE:i cromatidi si separano e s
i portano ai poli.
 700 LET a=14: LET b=16
 710 FOR j=1 TO 7
 720 PRINT AT 8,a; PAPER 4; INK
1;"\";AT 9,a;"/";AT 8,b; PAPER
 4; INK 1;"/";AT 9,b;"\": PRIN
T AT 12,a; PAPER 4; INK 7;"\";A
T 13,a;"/";AT 12,b; PAPER 4; IN
K 7;"/";AT 13,b;"\": BEEP .12,
3: LET a=a-1: LET b=b+1
 730 NEXT j
 740 PRINT INVERSE 1; INK 4;AT 1
1,13;"▀▀▀";AT 10,7;"▀";AT 10,23
;"▀"
 750 INVERSE 1: PLOT 60,92: DRAW
 125,0: PLOT 60,92: DRAW 125,0,-
PI/2: PLOT 60,92: DRAW 125,0,PI/
4: PLOT 60,92: DRAW 125,0,-PI/3.
5: PLOT 60,92: DRAW 125,0,PI/3.5
:
 760 INVERSE 0: PRINT AT 9,8; IN
K 1; PAPER 4;"/"
 770 INVERSE 1: PLOT 60,92: DRAW
 125,0,PI/2: INVERSE 0: PRINT AT
 9,8; PAPER 4; INK 1;"/"
 780 PRINT #0;AT 0,0; INK 9;"Ogn
i cromatidio ricostruisce l' alt
ro per riformare il cromosoma"
 790 PAUSE 150
 800 PRINT AT 8,9; PAPER 4; INK
1;"/";AT 9,9; PAPER 4; INK 1;"\"
;AT 12,9; PAPER 4; INK 7;"/";AT
13,9;"\"
 810 PRINT PAPER 4; INK 1;AT 8,2
3;"/"
 820 PRINT AT 8,22; PAPER 4; INK
 1;"\";AT 9,22;"/";AT 12,22; PAP
ER 4; INK 7;"\";AT 13,22;"/"
 830 GO SUB 1330: PRINT #0;AT 0,
0; INK 9;"TELOFASE:si ricostitui
scono i nuclei:la cellula si str
ozza in 2.": PAUSE 200:
 840 INK 9: CIRCLE 72,87,30: CIR
CLE 183,87,30: CIRCLE 72,87,32:
CIRCLE 183,87,32
 850 CIRCLE 55,87,5: CIRCLE 164,
87,5
 860 GO SUB 1330: PAUSE 50
 870 REM telofase
 880 LET y=175: INK 6
 890 FOR j=0 TO 87
 900 PLOT 115,j: DRAW 30,0
 910 PLOT 115,y-j: DRAW 30,0
 920 NEXT j
 930 REM 2 cellule
 940 PRINT AT 5,5; PAPER 4; INK
1;"▛";AT 6,5;"▜";AT 16,5; PAPER
4; INK 2;"▂▂";AT 5,9; PAPER 4; I
NK 7;"▟"
 950 PRINT AT 16,10; PAPER 4; IN
K 3;"▗";AT 2,4;"▗"
 960 PRINT AT 3,25; PAPER 4; INK
 3;"▗";AT 16,20;"▗";AT 16,23; IN
K 2;"▂▂";AT 2,20; PAPER 4; INK 1
;"▛";AT 3,20;"▜";AT 5,23; PAPER
4; INK 7;"▟"
 970 PRINT #0;AT 0,0; INK 9;"
                            "
 980 FOR j=.1 TO 3 STEP .1: BEEP
 .1,j: NEXT j
 990 STOP
1000 PAPER 0: INK 7: BORDER 7: C
LS : PRINT AT 10,7;"CELLULA E MI
TOSI.": PAUSE 100: BEEP .12,5: P
RINT AT 21,0;"Istruzioni?": PAUS
E 0:
1010 IF INKEY$="n" THEN PRINT AT
 21,0;"Attendere
": RETURN
1020 PAPER 7: INK 0: CLS
1030 PRINT AT 0,7;"Cellula e mit
osi."
1040 PRINT '''"Quando appare la
figura,premere un tasto per fare
 apparire le didascalie della fi
gura della cel-lula."''"Per la m
itosi,il procedimento   e' autom
atico.Per rivedere dare semplice
mente RUN.": PRINT AT 21,0;"Un t
asto per continuare": PAUSE 0
1050 PAPER 0: INK 0: BORDER 0: C
LS : RETURN
1060 FOR j=0 TO 7: READ a: POKE
USR "a"+j,a: NEXT j: DATA 24,60,
126,255,231,231,231,231
1070 FOR j=0 TO 7: READ a: POKE
USR "b"+j,a: NEXT j: DATA 231,23
1,231,231,255,126,60,24
1080 FOR j=0 TO 7: READ a: POKE
USR "c"+j,a: NEXT j: DATA 60,0,2
55,0,255,0,60,0
1090 FOR j=0 TO 7: READ a: POKE
USR "d"+j,a: NEXT j: DATA 63,224
,0,255,0,0,224,63
1100 FOR j=0 TO 7: READ a: POKE
USR "e"+j,a: NEXT j: DATA 252,7,
0,255,0,0,7,252
1110 FOR j=0 TO 7: READ a: POKE
USR "f"+j,a: NEXT j: DATA 16,84,
56,254,56,84,16,0
1120 FOR j=0 TO 7: READ a: POKE
USR "g"+j,a: NEXT j: DATA 0,96,2
48,252,127,63,15,7
1130 FOR j=0 TO 7: READ a: POKE
USR "h"+j,a: NEXT j: DATA 1,15,3
1,63,255,252,240,192
1140 FOR j=0 TO 7: READ a: POKE
USR "i"+j,a: NEXT j: DATA 128,24
0,248,255,255,63,31,7
1150 FOR j=0 TO 7: READ a: POKE
USR "l"+j,a: NEXT j: DATA 2,15,3
1,255,255,252,248,192
1160 FOR j=0 TO 7: READ a: POKE
USR "m"+j,a: NEXT j: DATA 16,16,
16,8,4,2,3,3
1170 FOR j=0 TO 7: READ a: POKE
USR "n"+j,a: NEXT j: DATA 3,3,2,
4,8,16,16,16
1180 FOR j=0 TO 7: READ a: POKE
USR "o"+j,a: NEXT j: DATA 8,8,8,
16,32,64,192,192
1190 FOR j=0 TO 7: READ a: POKE
USR "p"+j,a: NEXT j: DATA 192,19
2,64,32,16,8,8,8
1200 RETURN
1210 PLOT 50,20: DRAW 150,0: PLO
T 230,50: DRAW 0,90
1220 PLOT 200,20: DRAW 30,30,PI/
2
1230 PLOT 200,170: DRAW -150,0:
PLOT 200,170: DRAW 30,-30,-PI/2
1240 PLOT 20,50: DRAW 0,100
1250 PLOT 20,50: DRAW 30,-30,PI/
2: PLOT 20,140: DRAW 30,30,-PI/2
1260 RETURN
1270 INK 9: CIRCLE 128,87,30: CI
RCLE 128,87,32: PRINT AT 6,17; P
APER 4; INK 6;"*";AT 6,15; PAPER
 4; INK 6;"*"
1280 PRINT PAPER 4; INK 1;AT 9,1
5;"▀▀▀";AT 11,13;"▀▀▀";AT 13,1
4;"▀▀▀": INK 2: CIRCLE 110,93,5
1290 RETURN
1300 PRINT AT 8,15; PAPER 4; INK
 1;"\";AT 9,15; PAPER 4; INK 1;"
/";AT 8,16; PAPER 4; INK 1;"/";A
T 9,16; PAPER 4; INK 1;"\"
1310 PRINT AT 12,15; PAPER 4; IN
K 7;"\";AT 13,15; PAPER 4; INK 7
;"/";AT 12,16; PAPER 4; INK 7;"/
";AT 13,16;"\"
1320 RETURN
1330 BEEP .3,5: RETURN
```

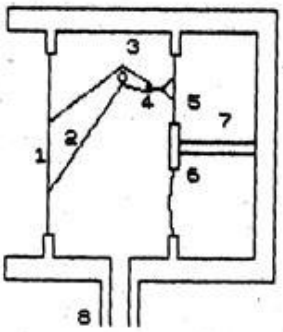
LISTATO B

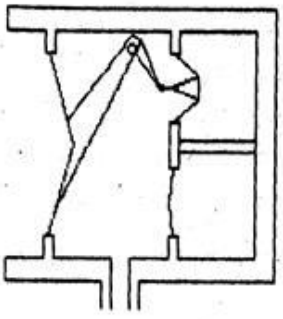
Autore: Paolo Coretti.
Nome del programma: Orecchio.
Linguaggio: Basic.
Hardware: Sinclair ZX SPECTRUM.
Scopo del programma: mostrare la struttura e il funzionamento dell'orecchio interno.

```
   2 REM © P. CORETTI - TRIESTE
   5 CLEAR 49999: CLS : BORDER 1
: PAPER 6: INK 2: CLS : PRINT AT
 10,10;"L'ORECCHIO": PAUSE 200
  10 BORDER 5: PAPER 7: INK 0: C
LS
  20 FOR j=58400 TO 58423: READ
a: POKE j,a: NEXT j: DATA 33,0,6
4,17,64,228,1,0,27,237,176,201,3
3,64,228,17,0,64,1,0,27,237,176,
201
  30 FOR j=50000 TO 50023: READ
a: POKE j,a: NEXT j: DATA 33,0,6
4,17,180,195,1,0,27,237,176,201,
33,160,195,17,0,64,1,0,27,237,17
6,201
 100 PLOT 90,130: DRAW 20,0: DRA
W 0,-10: DRAW 5,0: DRAW 0,10: DR
AW 60,0: DRAW 0,-10: DRAW 5,0: D
RAW 0,10: DRAW 40,0: DRAW 0,-100
: DRAW -40,0: DRAW 0,10: DRAW -5
,0: DRAW 0,-10: DRAW -20,0: DRAW
 0,-30
 110 PLOT 145,0: DRAW 0,30: DRAW
 -30,0: DRAW 0,10: DRAW -5,0: DR
AW 0,-10: DRAW -20,0: DRAW 0,-10
: DRAW 50,0: DRAW 0,-20
 120 PLOT 160,0: DRAW 0 ,20: DRA
W 70,0: DRAW 0,120: DRAW -140,0:
 DRAW 0,-10:
 130 PLOT 175,70: DRAW 5,0: DRAW
 0,20: DRAW -5,0: DRAW 0,-20: PL
OT 160,82: DRAW 40,0: PLOT 160,7
7: DRAW 40,0: PLOT 175,55: DRAW
2,15: PLOT 175,55: DRAW 2,-15
 145: RANDOMIZE USR 50000: GO SU
B 1600
 146 GO SUB 3000
 150 RANDOMIZE USR 58400
 160 RANDOMIZE USR 50012: GO SUB
 2000
 165 RANDOMIZE USR 50000
 170 PRINT #0; INK 9;"L'orecchio
 mentre percepisce un suono.": P
AUSE 200: CLS
 180 CLS
 200: FOR j=1 TO 30
 210 RANDOMIZE USR 58412:
 215 BEEP 1,1
 220 RANDOMIZE USR 50012
 230 NEXT j
1599 STOP
1600 PLOT 112,40: DRAW 0,80: PLO
T 177,120: DRAW 0,-30: PLOT 112,
59: DRAW 38,48: DRAW 15,-4: DRAW
 0,4: DRAW -15,8: DRAW -38,-25
1700 PLOT 170,105: DRAW 7,5: PLO
T 170,105: DRAW 7,-5: PLOT 171,1
05: DRAW -5,0: PLOT 169,105: DRA
W -5,0: INK 0: CIRCLE 150,110,2:
 CIRCLE 163,105,1
1701 RETURN
2000 PLOT 112,120: DRAW 13,-40:
DRAW -13,-40: PLOT 122,91: DRAW
33,37: DRAW 5,-2: DRAW -44,-74:
PLOT 160,125: DRAW 10,-22: PLOT
157,120: DRAW 13,-16
2010 INK 0: CIRCLE 155,122,2: CI
RCLE 170,105,1
2020 PLOT 170,105: DRAW 5,0: PLO
T 177,120: DRAW 13,-10: DRAW 0,-
10: DRAW -13,-10: PLOT 175,105:
DRAW 15,5: PLOT 175,105: DRAW 15
,-5
2100 RETURN
3000 PRINT AT 12,13;"1";AT 11,15
;"2";AT 6,19;"3";AT 9,20;"4";AT
9,23;"5";AT 13,23;"6";AT 10,25;"
7";AT 21,16;"8"
3020 FOR j=1 TO 8: READ z$: PRIN
T AT 0,0;z$: PAUSE 100: NEXT j:
DATA "1-Timpano","2-Martello","3
-Incudine","4-Staffa          ","5-Fi
nestra ovale","6-Finestra rotond
a","7-Columella          ","8-Tub
a di EUSTACHIO."
3030 PRINT AT 0,0;"
                  ";AT 12,13;" ";A
T 11,15;" ";AT 6,19;" ";AT 9,20;
" ";AT 9,23;" ";AT 13,23;" ";AT
10,25;" ";AT 21,16;" ";AT 9,20;"
  ";AT 10,25;"
4000 RETURN
```

L'ORECCHIO

2-Martello





LISTATO C

Autore: Giuseppe Bleiner.
Nome del programma: Leggi di Mendel.
Linguaggio: Basic.
Hardware: Sinclair ZX SPECTRUM.
Scopo del programma: mostrare le leggi di Mendel sui caratteri ereditari.

```
"
 100 PRINT AT 2,6;"UNITA' DIDATT
ICA N. 6";AT 10,9;"LEGGI DI MEND
EL";AT 20,6;"a cura di G. Bleine
r": PAUSE 200: CLS
 150 PRINT AT 2,13;"MENU'";AT 4,
6;"1  ISTRUZIONI";AT 5,6;"2  I L
EGGE DI MENDEL";AT 6,6;"3  II LE
GGE DI MENDEL";AT 7,6;"4  III LE
GGE DI MENDEL";AT 8,6;"5  INTERO
 PROGR.";AT 9,6;"6  USCITA"
 200 INPUT a: CLS
 205 IF a=1 THEN GO SUB 500
 210 IF a=2 THEN GO SUB 1000
 220 IF a=3 THEN GO SUB 1400
 230 IF a=4 THEN GO SUB 1780
 240 IF a=5 THEN GO SUB 500
 250 IF a=5 THEN GO SUB 1000
 260 IF a=5 THEN GO SUB 1400
 270 IF a=5 THEN GO SUB 1780
 285 IF a=6 THEN GO TO 2780
 287 GO SUB 2740
 290 GO TO 150
 500 PRINT "Un carattere eredita
rio per ma- nifestarsi deve esse
re dominan- te o se recessivo om
ozigote.    Si dice omozigote ri
spetto ad undeterminato caratter
e, un indi- viduo con una coppia
 di alleli  uguali, eterozigote
quello con  una coppia di alleli
 dissimili."
 520 PAUSE 400: CLS
 540 RETURN
1000 PRINT AT 2,5;"LEGGI DI MEND
EL"
1020 PRINT INVERSE 1;AT 4,3;"UNI
FORMITA' DELLA F 1"
1040 CIRCLE 48,98,11
1060 PRINT AT 9,5;"AA"
1080 CIRCLE 150,98,11
1100 PRINT AT 9,18;"bb"
1110 PRINT AT 9,22; FLASH 1;"gen
itori"
1115 PAUSE 80
1117 PRINT AT 9,22; FLASH 0;"gen
itori"
1120 FOR n=1 TO 8
1140 PAUSE 10
1150 PRINT AT 9,7+n;"->"
1180 PRINT AT 10,16-n;"<-"
1200 NEXT n
1220 FOR n=1 TO 4
1240 PAUSE 10
1260 PRINT AT 11+n,11;"!"
1280 PRINT AT 12+n,11;"v"
1300 NEXT n
1320 CIRCLE 94,22,12
1340 PRINT AT 19,11;"Ab";AT 19,1
6; FLASH 1;"F 1"
1360 PAUSE 100: PRINT ;AT 19,16;
 FLASH 0;"F 1"
1370 PRINT AT 21,0;"DOMINANTI (A
A), recessivi (bb)"
1380 PAUSE 300: CLS
1385 RETURN
1400 PRINT INVERSE 1;AT 4,1;"LEG
GE DELLA SEGREGAZIONE"
1420 CIRCLE 48,98,11: CIRCLE 150
,98,11
1440 PRINT AT 9,5;"Ab";AT 9,18;"
Ab": PRINT AT 9,25; FLASH 1;"F 1
"
1450 PAUSE 100: PRINT AT 9,25; F
LASH 0;"F 1"
1460 FOR n=1 TO 8
1480 PAUSE 15
1500 PRINT AT 9,7+n;"->"
1520 PRINT AT 10,16-n;"<-"
1540 NEXT n
1560 CIRCLE 23,40,12: CIRCLE 70,
40,12: CIRCLE 119,40,12: CIRCLE
168,40,12
1580 PRINT AT 16,2;"AA";AT 16,8;
"Ab";AT 16,14;"Ab";AT 16,20;"bb"
: PRINT AT 16,25; FLASH 1;"F 2"
1585 FOR n=1 TO 10 STEP 2
1600 PLOT 98,88: DRAW -6.5*n,-3.
0*n
1620 PLOT 98,88: DRAW 6.5*n,-3.0
*n
1640 PLOT 98,88: DRAW 2.2*n,-3.0
*n
1660 PLOT 98,88: DRAW -2.2*n,-3.
0*n
1662 PAUSE 10
1664 NEXT n
1666 PRINT AT 16,25; FLASH 0;"F
2"
1680 PRINT FLASH 1; INK 4;AT 20,
0;"[redacted]": PRINT AT
20,20;"25%"
1700 PRINT AT 21,8;"75%"
1720 PAUSE 200
1740 PRINT FLASH 0; INK 4;AT 20,
0;"[redacted]"
1760 PAUSE 300: CLS
1765 RETURN
1780 PRINT INVERSE 1;AT 2,0;"LEG
GE DI INDIP.ZA DEI CARATTERI"
1800 CIRCLE 128,128,11
1820 CIRCLE 68,128,11
1830 PRINT AT 6,5;"GG";AT 6,18;"
vv"
1835 PRINT AT 5,5;"LL";AT 5,18;"
rr"
1836 PRINT AT 6,23; FLASH 1;"Gen
itori"
1837 PAUSE 80
1840 PLOT 79,128: DRAW 19,-15
1860 PLOT 117,128: DRAW -19,-15
1870 PAUSE 80
1875 FOR n=0 TO 24 STEP 2
1880 PLOT 98,113: DRAW 0,-n
1882 PAUSE 10
1884 NEXT n
1960 PAUSE 100
1980 PRINT AT 6,23; FLASH 0;"Gen
itori"
2000 CIRCLE 98,78,11
2002 FOR n=1 TO 50 STEP 5
2004 PLOT 98,67: DRAW 0,-n
2006 PAUSE 10
2008 NEXT n
2020 PRINT AT 12,15;"Gv"
2040 PRINT AT 11,15;"Lr"
2060 PRINT AT 11,23; FLASH 1;"F
1"
2070 PRINT AT 18,20;"(segue)"
2075 PRINT AT 21,0;"DOMINANTI (G
,L), recessivi (v,r)"
2080 PAUSE 200
2100 PRINT AT 11,23; FLASH 0;"F
1"
2120 PAUSE 300: CLS
2140 PRINT INVERSE 1;AT 1,0;"LEG
GE DI INDIP.ZA DEI CARATTERI"
2160 PRINT AT 21,0;"DOMINANTI (G
,L), recessivi (v,r)"
2180 CIRCLE 118,138,11
2200 CIRCLE 58,138,11
2220 PRINT AT 5,3;"Gv";AT 5,18;"
Gv"
2240 PRINT AT 4,3;"Lr";AT 4,18;"
Lr"
2260 PRINT AT 4,27; FLASH 1;"F 1
"
2280 PAUSE 200
2300 PRINT AT 4,27; FLASH 0;"F 1
"
```

(segue a pag. 27)

(segue da pag. 26)

```
2310 FOR n=0 TO 10
2320 PLOT 69,128: DRAW 1.9*n,-n
2340 PLOT 107,128: DRAW -1.9*n,-
n
2342 PAUSE 10
2344 NEXT n
2360 PLOT 88,118: DRAW 0,-5
2380 PLOT 2,113: DRAW 170,0
2400 PLOT 10,113: DRAW 0,-10
2420 PLOT 60,113: DRAW 0,-10
2440 PLOT 110,113: DRAW 0,-10
2460 PLOT 160,113: DRAW 0,-10
2480 PRINT AT 10,0;"GGLL";AT 10,
18;"vvrr";AT 10,12;"Gvrr";AT 10,
6;"vvLr"
2500 PRINT AT 11,0;"GGLr";AT 11,
12;"GGrr";AT 11,6;"vvLL"
2520 PRINT AT 12,0;"GvLL";AT 12,
12;"Gvrr";AT 12,6;"vvLr"
2540 PRINT AT 13,0;"GvLr"
2560 PRINT AT 14,0;"GvLr"
2580 PRINT AT 15,0;"GvLr"
2600 PRINT AT 16,0;"GvLL"
2620 PRINT AT 17,0;"GGLr"
2640 PRINT AT 18,0;"GvLL"
2660 PRINT INVERSE 1;AT 18,14;"R
app. 9:3:3:1"
2680 PRINT AT 14,27; FLASH 1;"F
2"
2700 PAUSE 100
2720 PRINT AT 14,27; FLASH 0;"F
2"
2725 PAUSE 300: CLS
2735 RETURN
2740 FOR n=1 TO 6: BEEP .1,10: B
EEP .1,15: NEXT n
2745 RETURN
2750 PRINT AT 11,12;"F I N E"
```

UNITA' DIDATTICA N. 8

LEGGI DI MENDEL

a cura di G. Bleiner

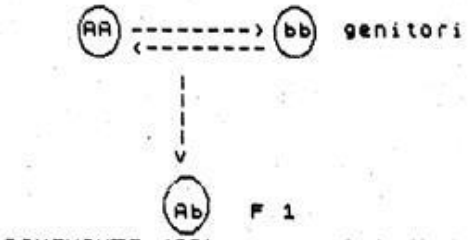
MENU'

1 ISTRUZIONI
2 I LEGGE DI MENDEL
3 II LEGGE DI MENDEL
4 III LEGGE DI MENDEL
5 INTERO PROGR.
6 USCITA

Un carattere ereditario per manifestarsi deve essere dominante o se recessivo omozigote. Si dice omozigote rispetto ad un determinato carattere, un individuo con una coppia di alleli uguali, eterozigote quello con una coppia di alleli dissimili.

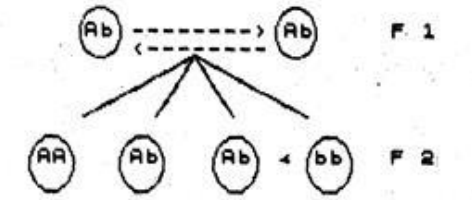
LEGGI DI MENDEL

UNIFORMITA' DELLA F 1

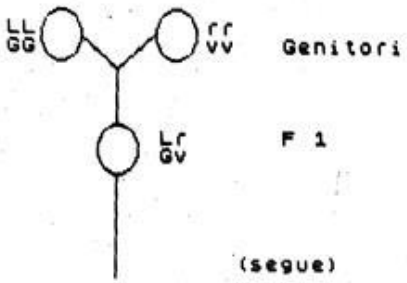


DOMINANTI (AA), recessivi (bb)

LEGGE DELLA SEGREGAZIONE



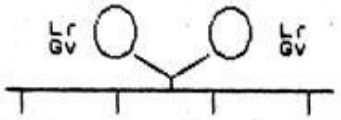
LEGGE DI INDIPENDENZA DEI CARATTERI



(segue)

DOMINANTI (G,L), recessivi (v,r)

LEGGE DI INDIPENDENZA DEI CARATTERI



GGLL vvLr Gvrr vvrr
GGLr vvLL GGrr
GvLL vvLr Gvrr
GvLr
GvLr F 2
GvLr
GvLL
GGLr
GvLL

DOMINANTI (G,L), recessivi (v,r)

LISTATO D

Autore: Paolo Coretti. La versione riprodotta è a cura di Giampiero di Dieco.
Nome del programma: Orecchio.
Linguaggio: Simons' Basic.
Hardware: Commodore 64.
Scopo del programma: mostrare la struttura e il funzionamento dell'orecchio interno.

```
1 REM *** RICHIEDE IL SIMONS' BASIC ***
2 REM *** BY GIAMPIERO DI DIECO ***
3 HIRES0,1
4 REM *** e PAOLO CORETTI ***
5 TEXT50,80,"L' ORECCHIO",1,5,20
6 TEXT110,180,"BY MEGASOFT",1,1,8
7 PAUSE8:PRINT"s":GOSUB1150
10 HIRES0,1:GOSUB20:GOTO480
15 REM *** DISEGNO STRUTTURA ORECCHIO **
*

20 LINE120,30,250,30,1
30 LINE250,30,250,130,1
40 LINE200,130,250,130,1
50 LINE200,130,200,150,1
60 LINE195,150,195,120,1
70 LINE195,150,200,150,1
80 LINE195,120,240,120,1
90 LINE180,150,185,150,1
100 LINE240,120,240,40,1
110 LINE120,40,240,40,1
120 LINE120,40,120,30,1
130 LINE120,120,185,120,1
140 LINE185,120,185,150,1
150 LINE180,150,180,130,1
160 LINE120,130,180,130,1
170 LINE120,130,120,120,1
180 LINE140,40,140,50,1
190 LINE140,40,145,40,0
200 LINE145,40,145,50,1
210 LINE140,120,140,110,1
220 LINE140,120,145,120,0
230 LINE145,120,145,110,1
240 LINE140,50,145,50,1
250 LINE140,110,145,110,1
260 LINE205,40,210,40,0
270 LINE205,40,205,50,1
280 LINE210,40,210,50,1
290 LINE205,50,210,50,1
300 LINE205,120,210,120,0
310 LINE205,120,205,110,1
320 LINE210,120,210,110,1
330 LINE205,110,210,110,1
340 LINE205,75,210,75,1
350 LINE205,75,205,95,1
360 LINE210,75,210,95,1
370 LINE205,95,210,95,1
380 LINE210,82,240,82,1
390 LINE210,87,240,87,1
400 A$="737737737737077070707"
410 ROT0,1:DRAWA$,207,95,1
420 PAINT125,35,1
430 PAINT125,125,1
440 PAINT209,90,1
450 PAINT219,83,1
460 PAUSE9
470 RETURN
475 REM *** MATRICI PER IL DISEGNO DELLE
 DUE POSIZIONI DEL MARTELLO ***
480 B$="77777777777777777777777777777777
77777777777777777777777777777"
490 C$="515515151551515155151551515151
551515155151515155151551515155"
500 C$=C$+"151515551507707077070770707
70770707777788618818181881886818181818"
510 C$=C$+"37377377373737373737377373737
37373737373737737373737373777"
520 C$=C$+"37373737373737373778"
530 D$="555551551551551551556666666677777
77777777777777777776666666688188"
540 D$=D$+"188188188188188888111111111111
13333333335550772888366 6"
550 E$="77707770777077077077077077077077
70777737773777377737773777377737777"
560 E$=E$+"3773777737777"
570 F$="506066060606060660606060606066066
06066560606060606665650707707 7"
580 F$=F$+"07757077070770770770707881818
18181818818181818377377377377"
590 F$=F$+"37377377373737737737737737737
73737787377373773738 8"
600 H$="55555555552555255552555525555255
55578282882828828828 8"
610 H$=H$+"00100101001010016666666666666
18188188181880022002002020028882888888"
620 H$=H$+"28888828888288828887"
630 DRAWB$,145,50,1
640 DRAWC$,145,75,1
650 DRAWD$,195,63,1
660 TEXT70,10,"LE PARTI DELL' ORECCHIO",
1,2,8
670 TEXT10,110,"R1.r - BTIMPANO",1,1,8:T
EXT135,80,"1",1,1,8
680 GOSUB1150
690 PAUSE3:TEXT10,120,"R2.r - BMARTELLO"
,1,1,8:TEXT152,75,"2",1,1,8
700 GOSUB1150
710 PAUSE3:TEXT10,130,"R3.r - BINCUDINE"
,1,1,8:TEXT195,45,"3",1,1,8
720 GOSUB1150
730 PAUSE3:TEXT10,140,"R4.r - BSTAFFA",1
,1,8:TEXT192,70,"4",1,1,8
740 GOSUB1150
750 PAUSE3:TEXT10,150,"R5.r - BFINESTRA''
OVALE",1,1,8:TEXT215,60,"5",1,1,8
760 GOSUB1150
770 PAUSE3:TEXT10,160,"R6.r - BFINESTRA
ROTONDA",1,1,8:TEXT212,100,"6",1,1,8
780 GOSUB1150
790 PAUSE3:TEXT10,170,"R7.r - BCOLUMELLA
",1,1,8:TEXT225,72,"7",1,1,8
800 GOSUB1150
810 PAUSE3:TEXT10,180,"R8.r - BTROMBA DI
 EUSTACHIO",1,1,8:TEXT187,155,"8",1,1,8
820 GOSUB1150:PAUSE10:GOSUB1150:GOSUB115
0
```

(segue a pag. 29)

(segue da pag. 28)

```
830 HIRES0,1
840 GOSUB20
850 TEXT10,170,"BL'ORECCHIO MENTRE PERCE
PISCE UN SUONO.",1,2,8
860 DRAWE$,146,50,1
870 DRAWF$,153,70,1
880 DRAWH$,190,62,1
890 TEXT80,70,"> > >",1,3,10
900 PAUSE8:PRINT"s":GOSUB1150:GOSUB1150:
GOSUB1150
910 HIRES0,1
920 GOSUB20
950 FORN=1TO30
960 DRAWB$,145,50,1
970 DRAWC$,145,75,1
980 DRAWD$,195,63,1
990 FORR=1TO50:NEXTR
1000 DRAWB$,145,50,0
1010 DRAWC$,145,75,0
1020 DRAWD$,195,63,0
1030 TEXT80,70,"> > >",1,3,10
1040 DRAWE$,146,50,1
1050 DRAWF$,153,70,1
1060 DRAWH$,190,62,1
1070 GOSUB1150
1080 DRAWE$,146,50,0
1090 DRAWF$,153,70,0
1100 DRAWH$,190,62,0
1110 TEXT80,70,"R      r",0,3,10
1120 NEXTN
1130 PAUSE9:PRINT"s":GOSUB1150:GOSUB1150
:GOSUB1150:GOSUB1150
1140 END
1145 REM *** BEEP ***
1150 POKE54296,15:POKE54295,0
1160 VV=54272:POKEVV+6,0:POKEVV+5,31
1170 POKEVV+1,180:POKEVV+4,33
1180 FORNN=1TO150:NEXTNN
1190 POKEVV+4,0
1200 RETURN

READY.
```

LISTATO E

Autori: Carlo Alberto Marchi, Emo Maracchia, Alessandro Migliarucci.
Nome del programma: Il fattoriale.
Linguaggio: Pascal.
Hardware: Univac 1100/82.
Scopo del programma: Calcolare e stampare il fattoriale di *n* numeri interi non negativi letti da scheda.

```
PROGRAMMA DI ESEMPIO PER LINGUAGGIO "PASCAL"


Il programma seguente si riferisce al compilatore PASCAL 1100
SRIA per elaboratore UNIVAC 1100/82 in servizio presso
L'Universita' di Roma.
Scopo del programma : Calcolare e stampare il fattoriale di n
numeri interi non negativi letti da scheda.
Descrizione dei tipi e delle variabili:
TIP0 A : tipo intero positivo.
TIP0 B : tipo intero non negativo.
MASSIMO : costante pari al massimo numero rappresentabile.
N : variabile di conteggio per ciclo di lettura e stampa.
NUMERO : numero di cui si vuole il fattoriale.
K : argomento della funzione FATT che calcola il fattoriale.
```

```
PROGRAMMA

PROGRAM FATTORIALE;
TYPE    TIP0A=1..MASSIMO;       (interi da 1 a massimo)
        TIP0B=0..MASSIMO;       (interi da 0 a massimo)
VAR     NUMERO:TIP0B;
           N:INTEGER;
           R:TIP0A;
FUNCTION FATT (K:TIP0B):TIP0A; (funz. ricorsiva che calcola il
                                        fattoriale)



BEGIN
  IF K=0 THEN FATT:=1      ( se k=0 il fattoriale e' 1)
         ELSE FATT:=K*FATT (K-1)
                           (altrimenti calcola il fattoriale)
END;



BEGIN                      (inizio del programma principale)
      READ (N);
      FOR N:=1 TO N
      DO BEGIN
            READ (NUMERO);
            WRITELN ("FATTORIALE DI",NUMERO,"=",FATT(NUM))
         END
END.
```

LISTATO F

Autori: M. Bonghi, M. Brandolini.
Nome del programma: Le macchine semplici.
Linguaggio: Pilot.
Hardware: Apple II.
Scopo del programma: mostrare con grafici e con animazione il principio della leva di 1° genere.

```
Print of lesson STAMPA

r: ****  AGGIORNAMENTO 19/9/84 *****

r:*** **** SET CARATTERI **** ***
r: altalena,italici,leva

r:*** **** VARIABILI **** ***
d:a$(30);s$(30)

r:*** **** SUBROUTINES **** ***
r:*quadro;cont


r:********* INIZIO LEZIONE *******
*leva
u:quadro
ts:v1,38,1,22;es
t:          UNITA' DIDATTICA n.1
ts:v;g1,6
ts:t2
t:          LE MACCHINE SEMPLICI:
ts:t1
ts:g1,12
ts:s2
t:       LA LEVA
ts:s1
ts:g1,20
t:          M.BONCHI e M.BRANDOLINI
r:
r:     tel. 06/5313435 - 0766/35085
u:cont


*1videata
ts:v1,38,1,22;es
ts:v1,38,2,22;12
t:     La leva e'costituita da un'asta
t:   rigida girevole su un punto fisso
th:           detto
ts:i
t:fulcro
ts:n
t: E'soggetta all'azione di due forze:
th:
ts:i
th:Potenza
ts:n
th: e
ts:i
t:Resistenza
ts:n
ts:v1,38,14,22;11
r:******* disegno leva tridimensionale
tx:leva
t:          vmmmmmmmmmmmmmmmmmx
t:          hjjjjjjjjfuljjjjjjjk
t:          ¦       cro       ¦
t:          ¦                 ¦
t:          g                 g
t:           P                 R
tx:
tx:italici
ts:v;g6,14
t:asta
ts:v;g18,17
t:fulcro
tx:
u:cont


*2videata
ts:v1,38,1,22;12;es
t:
t:Le leve a seconda della posizione del
t:        fulcro, si dividono in:
tx:italici
t:a) LEVA DI 1'GENERE O INTERFISSA
t:b) LEVA DI 2'GENERE O INTERRESISTENZE
t:c) LEVA DI 3'GENERE O INTERPOTENTE
tx:
w:3
t:
t:
th:Esaminiamo ora le leve di
tx:italici
t: 1'GENERE
tx:
u:cont


*3videata
ts:v1,38,1,22;es
tx:italici
th:
t:LA LEVA DI 1'GENERE ha il fulcro
th:
t:tra la potenza e la resistenza
tx:
t:ed e'in equilibrio quando la potenza
:P
t: e la resistenza R  sono inversamente
t:  proporzionali ai rispettivi bracci
t:
ts:t2
t:              P:R=b:a
ts:t1
ts:v1,38,15,22;11
r: ******* disegno leva
tx:leva
t:          <ttttttatt#tttbt>
t:          s--------F-----d
t:          ¦       ULC     ¦
t:         Pg                :R
t:                           ¦
t:                           g
tx:
u:cont


*4videata
ts:v1,38,4,11;es
```

(segue a pag. 31)

(segue da pag. 30)

```
t:
th:
ts:t2
t:e'vantaggiosa
ts:t1
t:
t:se il braccio della potenza
:e'maggiore
t:
t:      del braccio della resistenza
r: ANIMAZIONE OMINO
tx:omino
ts:v;g4,13;*6(aa/bc/de;afg/hi/jk;alm/nop
:/qr/;a.../.../..;wr;)
ts:g9,13;*1(aDAG/EBH/FC;)

ts:g36,13;*6(aas/tu/vw;axy/z0/12;a34/567/
:89;a../.../..;wl;)
ts:g28,13;*1(aDAG/EBH/FC;)

ts:g36,13;*4(aas/tu/vw;axy/z0/12;a34/56
:7/89;a../.../..;wl;)
ts:g31,13;*1(aDAG/EBH/FC;)
tx:
u:cont


*5videata
ts:v1,38,4,11;es
t:
th:
ts:t2
t:e'svantaggiosa
ts:t1
t:
t:se il braccio della potenza
:e'minore
t:
t:       del braccio della resistenza


r: ANIMAZIONE OMINO
tx:omino
ts:g31,13;*5(aa/bc/de;afg/hi/jk;alm/nop
:/qr;a../.../..;wr;)
ts:g2,13;*4(aa/bc/de;afg/hi/jk;alm/nqp/
:qr;a../.../..;wr;)
ts:g6,13;*1(aDAG/EBH/FC;)
tx:
u:cont


*esercizio
ts:v1,38,1,22;l1;es
ts:t2
t:                ESERCIZIO
ts:t1
t:
t:
t:  Due bambini giocano con l'altalena:
t:  uno pesa 30Kg e dista dal fulcro 3m
t:       l'altro dista dal fulcro 2m.
t: Calcolare il peso del secondo bambino
t: perche' l'altalena sia in  equilibrio
ts:v6,38,12,22
tx:altalena
```

```
ts:t2
t:      a                s
t:      bc              tu
t:      del111FUL1111111zh
ts:t1
t:      q      y         j
t:      q      v         j
t:      <xxxxxx#xxxxxxxxx>
t:      q  2m     3m     k
t:      q               30Kg
t:      w
t:      X
tx:
ts:v;g1,20
tx:italici
t:lavagna
tx:
r: ************ lavagna per conti
ts:i
ts:v1,10,14,19
t:
t:
t:
t:
t:
r: *********** routine per calcoli
ts:v1,10,14,19
a:$a$
c:s$="c:k="!!a$
xi:s$
t: #k
ts:n
ts:v;g10,22
th:Risposta :  X=
a:
r: ********* risposte considerate valide
m:%45!45kg!QUARANTACINQUE
ts:v1,10,14,21;es
tx:italici
r: ********* commento risposta esatta
ty:BENE.La
ty:risposta
ty:e'esatta
r: commento risposta errata
tn:errato
tn:ripassa
tn:la lezione
tx:
u:cont


r:************ SUBROUTINES **********
r: ***** cornice contorno videata
*quadro
g:v;m0,0;d0,511;d559,511;d559,0;d0,0
e:


r: ****** comando per cambio videata
*cont
ts:v
ts:g37,22
as:
m:C!c
th:
jn:cont
e:
```

LISTATO G

Autori: M. Bonghi, M. Brandolini.
Nome del programma: Geometria piana.
Linguaggio: Pascal.
Hardware: Apple II.
Scopo del programma: creare figure geometriche ed immagini ricorsive.

```
uses turtlegraphics;

const
larghezza=280;
altezza=192;
(* dimensioni pagina grafica *)



procedura assi;
begin
 pencolor(none);moveto(140,0);
 pencolor(white);moveto(140,190);
 pencolor(none);moveto(0,95);
 pencolor(white);moveto(280,95);
end;(* fine procedura assi *)



procedure quadrato (lato :integer);
var
i:integer;
begin
 pencolor(none);moveto (35,107);
 pencolor (green);
  for i :=1 to 4 do
   begin
    move (lato);
    turn (90);
   end
end;(*  fine procedura quadrato *)



procedure triangolo (lato : integer);
var
i:integer;
begin
 pencolor (none);moveto (175,107);
 pencolor (violet);
  for i := 1 to 3 do
   begin
    move (lato);
    turn (120);
   end
end;(* fine procedura triangolo *)



procedure esagono (lato : integer);
var
i:integer;
begin
 pencolor (none);moveto (50,10);
 pencolor (blue);
  for i := 1 to 6 do
   begin
    move (lato);
    turn (60);
   end
end;(* fine procedura esagono *)
```

```
procedure trapezio (lato : integer);
var
i : integer;
begin
 pencolor (none);moveto (160,20);
 pencolor (orange);
   move (lato);
   turn (120);
   move (lato-60);
   turn (60);
   move (lato-40);
   turn (60);
   move (lato-60);
end;(* fine procedura trapezio *)



 procedure figure (lato : integer);
 var
 i : integer;
 begin
   if lato > 10 then
     begin
      for i := 1 to 4 do
       begin
        move (lato);
        turn (90);
       end; (* for *)
      move (lato div 2);
      turn (45);
      figure (trunc (lato*0.707))
   end (* if *)
 end;(* fine procedura figure *)



 procedure ricorsione (lato : integer);
 var
 i : integer;
 begin
  if lato >1 then
   begin
    for i := 1 to 4 do
     begin
      move (lato);
      turn (90);
    ricorsione ( trunc (lato*0.45))
    end;
  end (* if *)
 end;



(* programma principale *)
begin
 initturtle;
 assi;
 quadrato(70);(* 70 = lato quadrato *)
 triangolo(70);(* 70 = lato triangolo *)
 esagono(40);(* 40 = lato esagono *)
 trapezio(100);(* 100 = base maggiore trapezio *)
 readln;
 initturtle;
 pencolor (none);moveto(44,0);
 pencolor (white);
 figure (191);(* 191 = lato figura esterna *)
 readln;
 initturtle;
 pencolor(none);moveto(44,0);
 pencolor(white);
 ricorsione(191);(* 191 = lato cornice *)
 readln;
 textmode
end.
```

BIBLIOGRAFIA

È stata consultata la collezione completa delle seguenti riviste di informatica: « Micro & Personal Computer », « Microcomputer » e « Bit ».

J. PIAGET, *Logique et connaissance scientifique*, in « Enciclopedie de la Pleiade », Gallimard, Parigi 1967.

F. HELMAR, *Pedagogia e cibernetica*, Armando Armando, Roma 1974 (edizione italiana a cura di G. Lariccia).

A. RALSTON, *Introduction to Programming and Computer Science*, Mc Graw Hill, 1971.

A. J. T. COLIN, *Fundamental of Computer Science*, Mc Millan, 1978.

J. L. TAYLOR - R. WALFORD, *I giochi di simulazione*, a cura della Open University, Mondadori, Cles 1979.

Open University, *Pensare per modelli*, Mondadori, Cles 1980.

G. LARICCIA, *Le radici dell'informatica*, Sansoni, Bologna 1981.

A. FRENKEL, *Introduzione all'uso dei piccoli calcolatori: struttura, programmi, applicazioni*, La Nuova Italia Scientifica, Urbino 1982.

G. ALBERTINI, *Introduzione alla pratica del sistema operativo*, suppl. al n. 38 di « Bit » (apr. 1983).

Scuola e computer, numero speciale di « Media Duemila », n. 2 (ott. 1983).

B. S. GOTTFRIED, *Programmare in Basic*, Etas, Farigliano, 1983.

R. RASCHETTI - R. MELCHIORI - C. OCCHIONERO, *Computer Due*, Signorelli, Roma 1982.

R. MATEOSIAN, *Giocare con il Basic*, Jackson, Milano 1982.

T. DWYER - M. CRITCHFIELD, *Il Basic e il Personal Computer. Uno: introduzione*, Muzzio, Trento 1983.

T. DWYER - M. CRITCHFIELD, *Il Basic e il Personal Computer. Due: applicazioni*, Muzzio, Trento 1983.

V. FALZONE - G. POMPEI, *Elaboratori elettronici e loro applicazioni*, I, Calderini, Bologna 1983.

S. PAPERT, *Mindstorms: bambini, computer e creatività*, Emme, Milano 1983.

R. DIDDAY, *Intervista sul Personal Computer: Software*, Muzzio, Trento 1983.

La rete italiana di documentazione pedagogica, a cura della Biblioteca di Documentazione Pedagogica, Firenze 1984.

S. UBALDI - G. RADIVO - C. BIAGINI, *Il Computer nella scuola di base*, Coletti, Tivoli 1984.

M. LAENG - G. LARICCIA - J. MEGARRY - J. P. SIMON - T. O'SHEA - G. FISCHER - G. MAURI - M. ALBERTI, *I ragazzi e il computer, una sfida per l'editoria*, volumetto presentato in occasione della Fiera del libro per ragazzi, Bologna 1984.

D. P. BOVET - G. CIONI, *Il mondo dei Personal Computer*, La Nuova Italia Scientifica, Urbino 1984.

Dizionario di telecomunicazione e telematica, a cura della rivista « Zerouno » e della ITALTEL, v. I, II, Cologno Monzese 1984.

E. PENTIRARO, *A scuola con il computer*, Laterza, Bari 1984.

M. LAENG, *L'educazione nella civiltà tecnologica*, Armando, Roma 1984.

H. C. REGGINI, *Logo: ali per la mente*, edizione italiana a cura di Giovanni Lariccia, Edizioni Elettroniche Mondadori, Milano 1984.