

PROGRAMMI PER IL TUO HOME COMPUTER

SISTEMI MSX
AUTOPROGRAMMA
grafico



TELEMATICA

LA STRUTTURA DI
AMIGA

SISTEMA OPERATIVO
MS-DOS

All'interno: SCHEDE TECNICHE



OLIVETTI PRODEST PC1



COMMODORE PC1

UN PERSONAL
COMPUTER
PER TUTTI

smal'87

PROGRAMMI
PROGRAMMI
PROGRAMMI

olivetti

PRODEST

COMMODORE

SISTEMI MSX

ZX-SPECTRUM

SHARP MZ 700



MULTITECH



ATARI PC

INTELLIGENZA ARTIFICIALE
...IL TURBO-PROLOG

All'interno: **CARTOLINA IDENTIKIT**

Tutti i segreti che bisogna conoscere per acquistare bene Computer, Monitor, Videogiochi.

In un mercato effervescente come quello dell'elettronica, non è facile orientarsi tra tante offerte tutte diverse ed accattivanti. ■■ Acquistare bene a colpo sicuro, non è impossibile: l'importante è avere ben presenti le proprie esigenze, senza farsi distrarre da tanti inutili "gadgets" che non aiutano certo la qualità. Anzi, tanto più una macchina è semplice, tanto più è affidabile e sicura. ■■ Una macchina semplice, tuttavia, non deve essere per questo limitata, ma deve consentire una certa possibilità di espansione e deve essere aperta ad accogliere nuovi programmi non ancora realizzati. ■■ Esattamente come la linea computer Fenner: espandibile secondo le le Vostre esigenze, capace di gestire video ad alta risoluzione, in grado di fornire delle sofisticate macchine di calcolo o divertenti partners di video giochi. ■■



In una parola.



FENNER®

Un grande nome nell'elettronica

Raccomandiamo la brevità delle domande. Si darà precedenza a quelle che trattano argomenti di interesse generale.

OLIVETTI PRODEST E PC1

Possiedo un PC-128 S Olivetti e sono soddisfatto delle sue potenzialità, manca però una valida e completa biblioteca software.

Venuto a conoscenza del nuovo prodotto della Olivetti-Prodest: il PC1, mi sorge il dubbio che il mio PC-128 S sia divenuto ormai obsoleto e privo di «potenzialità future»; il PC1 è compatibile quindi come voi stessi scrivete con una inesauribile biblioteca software, senza contare delle prevedibili scelte della stessa Olivetti Prodest in materia.

Altra cosa che mi interessa sapere è la giusta stampante per il mio «vecchio» PC-128 S. Non sfornerei listati giornalmente, e quindi non ho necessità di alte capacità tecniche né di lunghe operazioni di scrittura.

Formigatti Angelo - Conselice (RA)

Dire che il PC-128 S è vecchio è forse un'affermazione un po' esagerata, il suo è un dubbio che ultimamente assilla un gran numero di possessori di PC-128 S, dopo la presentazione del PC1. Ragazzi, rimanete tranquilli, il PC1 non è un'evoluzione tecnologica del 128 S, bensì un compatibile con lo standard MS-DOS (tranne che per il formato dei dischi) quindi una macchina di tutt'altra fascia; inoltre i due modelli Prodest (PC-128 e PC-128 S) sono ancora presenti sul catalogo e di conseguenza ancora costruiti senza il rischio di intralciare il passo al PC1 e viceversa. Per quanto riguarda il software, la casa mette a disposizione dei programmi base (WP, Data Base, ecc.), anche se da molti negativamente commentati, comunque è ormai imminente l'uscita sul mercato della «guida di riferimento» dalla quale si potranno attingere interessanti informazioni sulla macchina.

Conclusione? Studiando bene il calcolatore (e i mezzi ci sono), il software potrai svilupparlo da solo come fanno i nostri collaboratori programmatori. La stampante per PC-128 S è la DM-90 S Olivetti Prodest, ed è ottima.

PROPOSTE PER IL PC-128

Sono un miserrimo utente del PC-128 Olivetti Prodest e vorrei dirvi alcune cose sperando di non annoiarvi:

1) leggendo LIST talvolta ho incontrato dei bei programmi per altri computers mi sono cimentato nella loro traduzione. I lin-

guaggi che più mi sono comprensibili sono gli MSX e lo Sharp MZ 700, nonché lo Spectrum. Degli altri computers, (vedi C-64, C-128, ecc.) mi riesce praticamente impossibile alcuna decifrazione per l'elevato numero di peek e poke di cui non conosco la mappa di memoria. Vorrei dunque proporvi di mettere accanto ad un listato (ad esempio per lo Sharp MZ 700), delle note esplicative che rendano agevole la traduzione da un tipo di Basic ad un altro;

2) Vorrei che dedicaste una rubrica (vedi «Di Tutto un po'» per il mio calcolatore. So che gli Olivetti PC-128 sono abbastanza diffusi per cui accontentando me, accontentereste una vasta gamma di lettori (nella mia classe ho diffuso la vostra rivista; ora i «LISTIANI» sono 10 in più!!);

3) Se dedicare una rubrica vi sembra eccessivo, potreste almeno proporre delle utility che si adattino ad un pubblico giovane che va dalle scuole medie fino all'università e proporre argomenti matematici, fisici, biologici, chimici.

Potrete mai accontentarmi?

Vi scongiuro.

Roberto Picco - Salice (PN)

Non preoccuparti, se rispondere alle richieste dei lettori fosse una noia, non esisterebbe la rubrica, quindi procediamo con ordine (1, 2, 3) con le tue richieste.

1) È ammirabile il tuo impegno nel tradurre i programmi che trovi sulla rivista nel Basic del PC-128, e diciamo comprensibile la richiesta di allegare ad ogni listato le conversioni per altre macchine. In verità abbiamo valutato questa possibilità ma se veramente per ogni programma dovessimo effettuare un lavoro di adattamento per altri computers (cosa che poi dovremmo fare molto bene e non alla buona) troveresti LIST in edicola ogni 3 mesi.

2) Sulle pagine di LIST (nella rubrica «Per saperne di più...») iniziano ad affiorare quelli che vengono chiamati trucchi, routine anche per la serie Prodest in genere e questo sappiamo che farà piacere a tutti gli utenti Olivetti.

3) Per quanto riguarda gli argomenti che tu ci proponi, be' considera che dei nostri programmatori Olivetti, due sono studenti universitari iscritti a facoltà scientifiche, un altro è biologo quindi profondi conoscitori delle discipline che citi nella richiesta 3.

Concludiamo assicurandoti che su LIST ci sarà un costante appoggio a tutti gli utenti del PC-128 e che le tue richieste se non sono state già esaudite, lo saranno al più presto.

SPHEROGRAPH PC-128 S

Vi scrivo per segnalarvi una correzione da apportare al programma «Spherograph» per PC-128 S (comparso su LIST n. 3/1987).

Il programma «Spherograph», in fase di listato pubblicato funziona egreggiamente (complimenti agli autori!), ma per una svista nella stampa del listato non esegue le rotazioni intorno ai 3 assi X, Y e Z.

Per far sì che esegua dette rotazioni occorre correggere le linee 810 ed 820 del listato, sostituendo in esse X3, Y3, e Z3 ed X, Y, Z (se no il programma calcola sì la rotazione di coordinate, sino ad ottenere X3, Y3, Z3, ma poi non le applica nella trasformazione finale, appunto nelle linee 810 ed 820!!!).

Cordiali saluti e buon lavoro con la vostra bella rivista!

Gaudenzio Villani - Cambiano (TO)

Abbiamo ricevuto la tua lettera nella quale ci hai fatto notare la presenza di un piccolo «bug» nel programma «Spherograph» per PC-128 S, pubblicato sul numero di marzo. In effetti hai ragione: a causa di una nostra svista, abbiamo pubblicato un listato modificato in fase di progettazione, e quindi concettualmente errato. Ci scusiamo con i lettori di questo inconveniente, peraltro poco grave. Infatti l'errore si presenta solo quando si vuole ruotare il grafico attorno ad un asse, operazione che non si effettua molto frequentemente. Ti ringraziamo ancora una volta per il tuo interessamento, e ti salutiamo invitandoti a collaborare ancora con noi.

ERRATA CORRIGE

Nel numero 8/9 u.s. di questa rivista, a pag. 24 riga 16^a leggasì IBM anziché ISM; a pag. 25 riga 14^a leggasì PDP anziché Pop; a pag. 64 listato «OLI3D» nella riga 820, l'ultimo numero di riga è 3560, e quello della riga 840 è 800.

Ci scusiamo con i nostri lettori.

AVVISO

La pagina degli annunci sarà pubblicata nel prossimo numero

SOMMARIO *list* 10

RUBRICHE

4 LA POSTA DEI LETTORI

8 EDITORIALE

11 NOTIZIARIO

18 TELEMATICA

20 RECENSIONI LIBRI

22 MS-DOS

28 INTELLIG. ARTIFICIALE

31 PER SAPERNE DI PIÙ...

43 TECNICHE DI COMPUTERGRAFICA

81 SCUOLA E COMPUTER

87 SERVIZIO PROGRAMMI



PER SAPERNE DI PIÙ...

Siamo sempre più convinti che l'utilità di una rivista come LIST sia ottenibile con la pubblicazione di rubriche che formino ed amplino il bagaglio culturale dei lettori nello specifico settore dell'informatica e della computeristica.

Oltre ai «programmi pronti» per i computers più diffusi, che ci sforziamo di scegliere e presentare sempre più interessanti e divertenti, ecco che, proprio da questo numero, viene sostituita la rubrica «Di tutto un po'» con «Per saperne di più...» che allargherà le conoscenze computeristiche avvalendosi della collaborazione di esperti, ma aperta anche alle esperienze dei lettori, «Per saperne di più...» è un altro nuovo componente di quella originale formula editoriale che si chiama «LIST».

PROGRAMMI PER HOME-COMPUTER

38 SHARP SPAZIO POCKET

40 SHARP MZ 700 TOTOSHARP

47 SISTEMI MSX AUTOPROGRAMMA GRAFICO

54 OLIVETTI PRODEST PC128 PIE CHART

58 OLIVETTI PRODEST PC128 SQUIGGLE

60 OLIV. PRODEST PC128S AGENDA APPUNTAMENTI

65 COMMODORE C-128 TUTTO DOS

68 COMMODORE C-64 MULTITASK

70 COMMODORE C-64 CRAYON

72 COMMODORE C-64 PUZZLE

75 ZX SPECTRUM FAST LANDER

77 ZX SPECTRUM DEC HES

LIST PROGRAMMI PER IL TUO HOME COMPUTER — È UNA PUBBLICAZIONE DELLA EDICOMP S.R.L. - VIA F. STILICONE, 111 - 00175 ROMA

DIRETTORE EDITORIALE E RESPONSABILE: Renzo Rubeo — **COORDINATORE TECNICO** Fabrizio Nicolai — *Hanno collaborato a questo numero:* Paolo Ciancarini, Alessandro Ceracchi, Angelo Coccettini, Osvaldo Contenti, Maurizio Erban, Enrico Fabrizi, Roberto Feltoni, Giuseppe Festa, Marco Marzano, Fabrizio Nicolai, Elisabetta Portacci, Gian Piero Rosi, Fabio Russo, Fabio Scafati, Giulio Vannini, Marcello Zerbini —

DIREZIONE, REDAZIONE, AMMINISTRAZIONE E PUBBLICITÀ: Via F. Stilicone, 111 — 00175 Roma — Tel. 06/7665094 — **FOTOCOMPOSIZIONE:** GRAPH-SYSTEM di L. Pagano, s.n.c. — Via Gaio Melisso, 47 — 00175 Roma — Tel. 06/763175 — **STAMPA:** Fratelli Spada — V. Lucrezia Romana 00043 Ciampino/Roma — **DISTRIBUTORE ESCLUSIVO:** Messaggerie Periodici — Viale Famagosta, 75 — 20141 Milano — Tel. 02/8466545.

Registrazione e Autorizzazione presso il Tribunale di Roma, n. 254 del 3/8/1983 — Spedizione in abbon. post. gruppo III — 70%.

Prezzo di un numero: L. 5.000 — Numero arretrato: L. 7.000 — Abbonamento annuo: L. 55.000. Per l'estero: L. 110.000. I pagamenti vanno effettuati a mezzo c.c. bancario, vaglia postale, c.c. postale n. 72609001 intestato a LIST programmi per il tuo home computer. Casella postale 4092 ROMA APPIO. Per i cambi di indirizzo allegare alla comunicazione l'importo di L. 500, anche in francobolli, e indicare insieme al nuovo anche il vecchio indirizzo. Tutti i diritti di riproduzione, anche parziale, del materiale pubblicato sono riservati. Manoscritti, listati, bozzetti e fotografie anche se non pubblicati non si restituiscono. La Direzione declina ogni responsabilità in merito alla originalità, alla provenienza e alla proprietà dei programmi pubblicati. La responsabilità dei testi e delle immagini pubblicate è imputabile ai soli autori. L'invio di materiale, da parte degli autori, implica il consenso alla pubblicazione. Qualunque citazione di prodotto, marca, indirizzo è data solo a titolo informativo senza scopo pubblicitario, per l'unica documentazione del lettore. Per ogni controversia è competente il Foro di Roma.

Copyright by Edicomp s.r.l.

STAMPATO IN ROTOFFSET



Associato all'USPI
Unione Stampa
Periodica Italiana.

PC 1. TUTTO QUEL AI COMPUTE

NUOVO.

PC 1 è il nuovo grande computer della Olivetti Prodest. Il computer per casa per futuri numeri uno.

POTENTE.

512 K RAM espandibili fino a 640 Kbyte tramite scheda d'espansione. Processore a 4.77 / 8 Mhz. Utilizza dischi da 3,5" in grado di contenere fino a 720.000 informazioni ed è predisposto per un secondo drive da 3,5" che raddoppia la sua capacità di gestione.

COMPATTO.

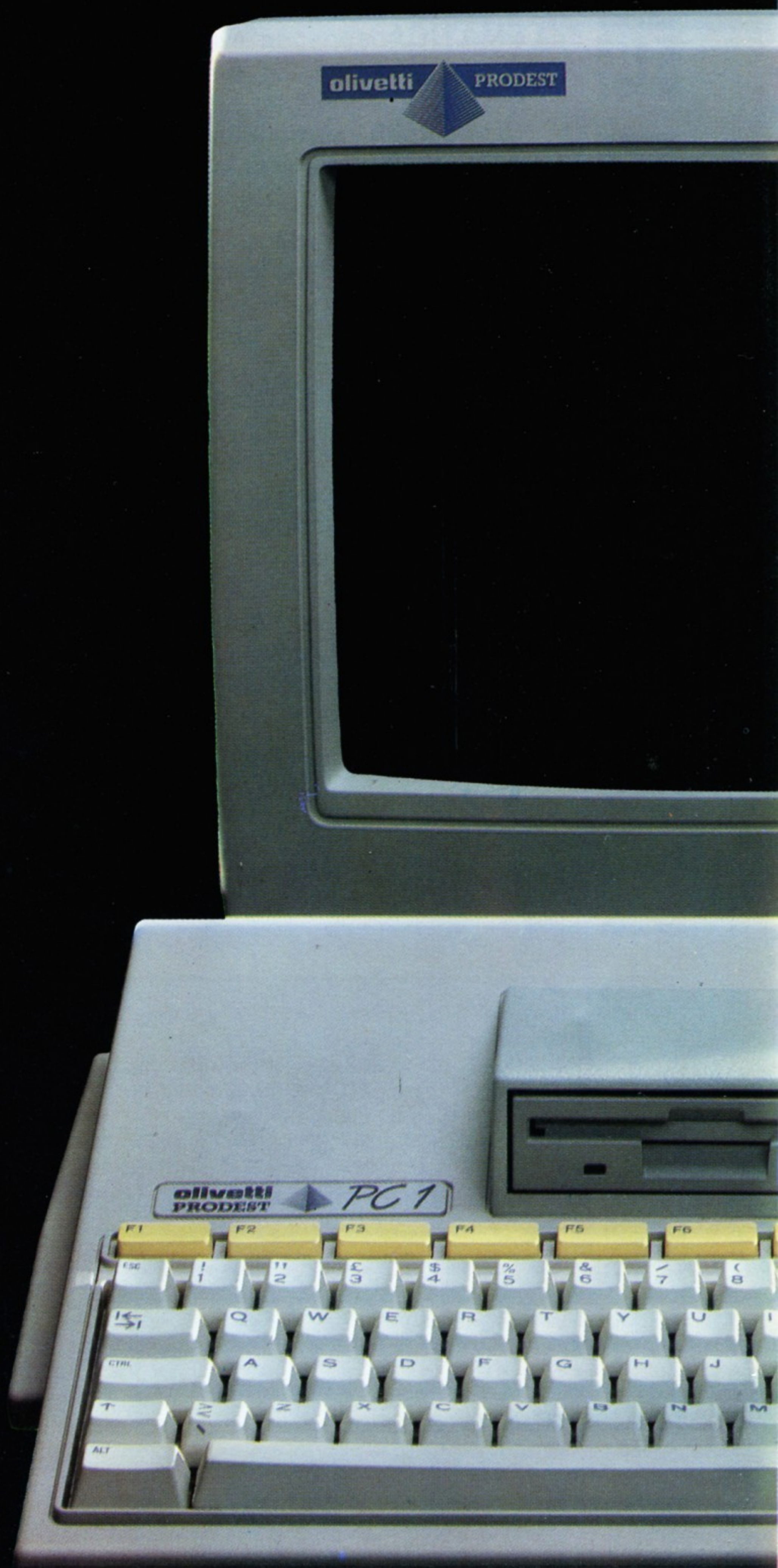
Grazie alla sua architettura geniale e sofisticata e al design d'avanguardia il PC 1 è compatto e completamente integrato. Tutte le funzioni sono infatti racchiuse in due soli pezzi: tastiera con drive incorporato e monitor.

COMPLETO.

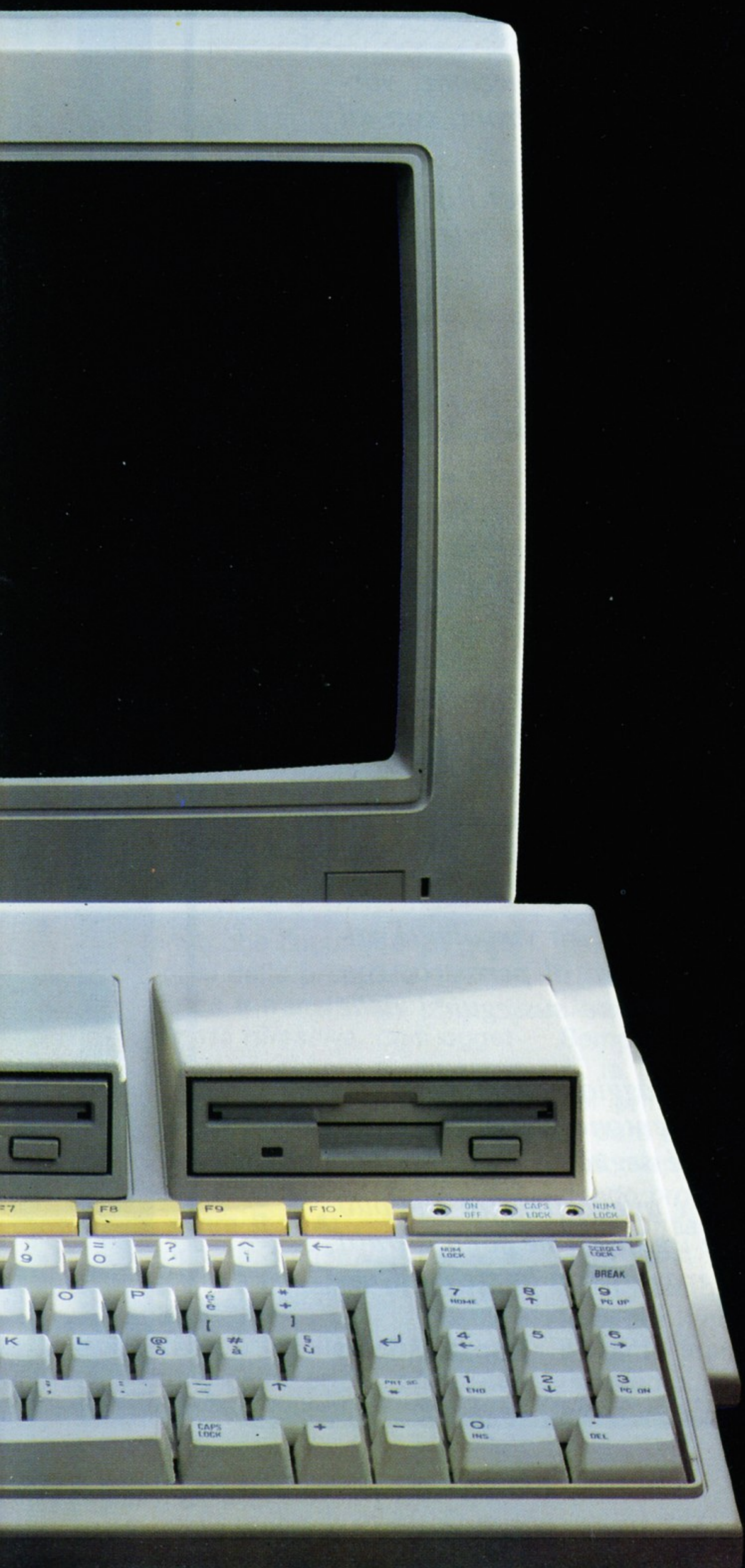
Il PC 1 è completo di tutto. Tastiera Qwerty a 83 tasti, con sezione alfanumerica, funzioni (10) e pad numerico; drive da 3,5" incorporato. Monitor monocromatico a fosfori verdi da 12". Disco di sistema: MS/DOS®, GW® BASIC, LODERUNNER GAME. "Per cominciare subito", un package di programmi che permette di utilizzare subito il PC 1 (WP, Grafica, Data tree, Tutorial e Totocalcio).

COMPATIBILE.

Il PC 1 è IBM® compatibile. Utilizza infatti il sistema operativo



LO CHE MANCAVA R PER CASA.



MS/DOS® 3.2, il linguaggio standard utilizzato dai personal computer professionali della nuova generazione.

ESPANDIBILE.

Il PC 1 non ha limiti. La sua ricca dotazione di periferiche può soddisfare qualsiasi esigenza: stampante, drive da 5,25", joystick, mouse, monitor a colori. E grazie al Bus di Espansione, IBM® compatibile, è possibile collegare al PC 1 i principali hardware presenti sul mercato.

DOTATO.

Una dotazione di software che nessun altro può vantare. Programmi di produttività personale, videogiochi e grafica, tutti di altissimo livello e a prezzi sbalorditivi. E grazie alla compatibilità di linguaggio è possibile utilizzare anche i programmi professionali delle principali software-house del mondo sviluppati in ambiente MS/DOS®.

1.195.000

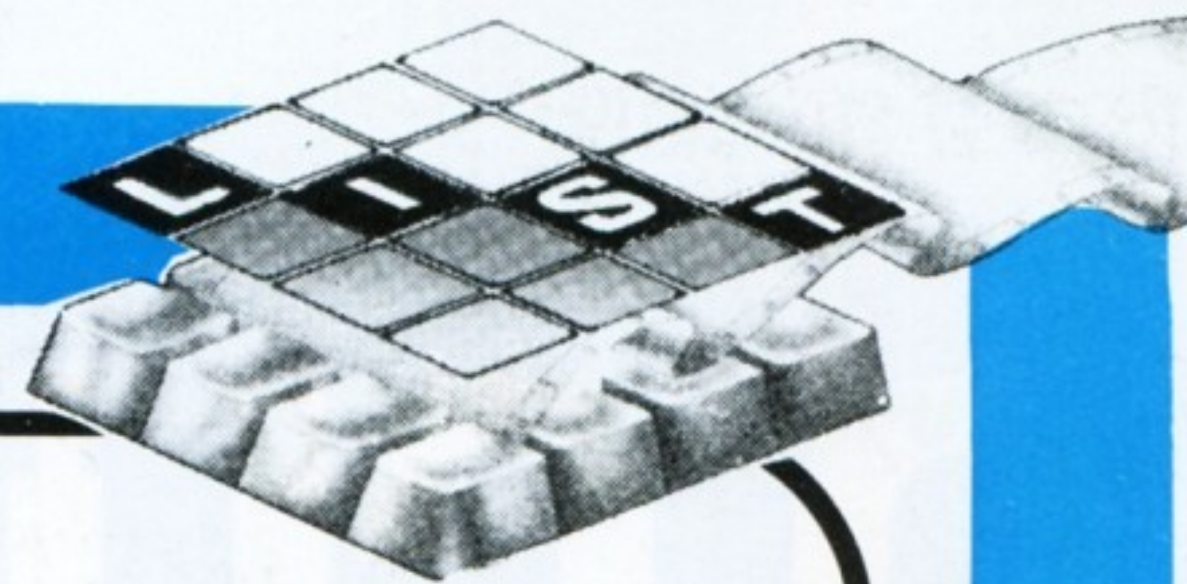
512 Kbyte, un floppy da 3,5", monitor monocromatico, sistema operativo MS/DOS®, linguaggio GW® BASIC, LODERUNNER GAME. Più IVA.

olivetti

PRODEST



I COMPUTER PER CASA.



Musica leggera e computer

In una nota — sempre lucida e interessante — del sociologo Francesco Alberoni, pubblicata di recente sul *Corriere della Sera*, abbiamo letto che «gli adolescenti, per esprimere la loro specificità e diversità, devono riunirsi in gruppo, formare una società in cui siano possibili scambi di esperienze. Devono elaborare un proprio linguaggio... un gergo. E — prosegue Alberoni — l'unico linguaggio specificamente loro, diverso da quello delle generazioni precedenti, è la musica leggera».

D'accordissimo, concordiamo noi, ma vorremmo sollevare una obiezione: vorremmo cioè sostituire quell'«unico linguaggio», dichiarato da Alberoni, con «il più diffuso».

E chiariamo: come dimenticare che oltre la musica leggera esiste per il giovane d'oggi quello strumento, giovane tra i giovani che si chiama «computer»?

Il computer rappresenta il nuovo gergo... il nuovo «look», anche intellettuale, che affascina l'adolescente prima, con i suoi videogiochi, e il giovane poi, con il suo linguaggio autenticamente moderno e futuribile.

E l'entusiasmo che il computer suscita nel giovane è paragonabile — nel senso sociologico — a quello per la musica leggera. Diventa il suo specifico, lo accomuna agli altri giovani nello scambio di esperienze e di scoperte.

Apparentemente il computer sembra isolare il giovane davanti al monitor; in realtà sperimentare quello strumento lo rende membro della comunità modernissima di tutti quei giovani che, come lui, lo stanno sperimentando e delle cui esperienze poi parleranno, discuteranno, su cui si confronteranno e... socializzeranno.

Dunque il computer, come e a volte più della musica leggera, diventa specificamente loro, li distingue e li esalta (pur senza raggiungere il fanatismo). Li fa riunire in gruppi, in clubs (oltre 1.500 nella sola Italia).

Proprio perché linguaggio di grandissima attualità e di sicura proiezione nel futuro, la computeristica è, con la musica leggera, l'altro aspetto di identificazione dei giovani.

Come la musica leggera, la computeristica ha una sua struttura vincolante eppure libera, capace ad esempio di dare immediatezza grafica al pensiero, con ritmi ed immagini paragonabili, guarda caso, all'altalenante susseguirsi delle note musicali...

Ecco perché, secondo noi, esiste un altro specifico linguaggio nella società dei giovani. Essi si interessano e credono nell'informatica; sentono che è una disciplina che crescerà con loro; ne sono militanti ancorché seguaci.

Il computer è il loro e nostro nuovo credo; il monitor è quella finestra che può offrire sempre più vasti orizzonti, ed è per questo che i giovani ne sono attratti e a quella finestra appassionatamente si affacciano... magari con lo stereo acceso a tutto volume.

Renzo Rubeo

Servizio **NOVITÀ
ESCLUSIVO**
*per tutti coloro che hanno
inviato o che invieranno subito
la **CARTOLINA-IDENTIKIT***

GRATIS
AL TUO DOMICILIO
RICEVERAI PERIODICAMENTE

*una
speciale
BUSTA
personalizzata
contenente:*

**OPUSCOLI
CATALOGHI
HOUSE-ORGANS
PIEGHEVOLI
CIRCOLARI
OFFERTE SPECIALI
PROMOZIONALI ecc. ecc...**



*delle migliori Case Produttrici internazionali
di computer - accessori - periferiche e software*

**Certamente ti interessa essere
sempre tra i più informati !
Se ancora non hai inviato la
CARTOLINA-IDENTIKIT
fallo subito ! La trovi allegata
nelle pagine centrali già affrancata**

**LIST MAILING
INFORMATICA**
un servizio per te !

Storia del **Biometal®**

La ricerca scientifica ci ha costantemente stupito per scoperte affascinanti, progressi in ogni direzione, sforzi per raggiungere risultati sempre più precisi sia nel mondo macroscopico, sia in quello microscopico.

Quando l'effetto transistor fu studiato in un laboratorio si intuì la portata di una scoperta che è ad oggi la base della tecnologia e, ormai, della vita comune.

Anche se quella scoperta fu così rivoluzionaria chissà quanti lettori di riviste specializzate o inserti scientifici in quotidiani, avranno appreso la notizia e poi, dopo un classico «ohh! incredibile» avranno proseguito in altre letture di contenuto più attuale e concreto (per quei tempi naturalmente).

Potrebbe essere il caso di questo articolo, cari lettori, considerata la novità di un prodotto comparso da pochissimo sul mercato mondiale: **BIOMETAL®**.

BIOMETAL® è una lega titanio-nichel che possiede incredibili capacità di modificare la propria struttura cristallina al variare della temperatura; fenomeno che prende il nome di SME (Sharp memory effect).

Per chiarire meglio cosa sia l'effetto SME pensate ad un pezzo di filo metallico che misuri 10 cm e che si accorci di 1 cm se riscaldato e, viceversa, si allunghi di 1 cm se raffreddato.

Per valutare l'utilità di tale scoperta è opportuno ripercorrere le sequenze più significative degli studi sulle leghe ad effetto SME, evoluzioni che nel corso di 40 anni hanno reso possibile il passaggio da ricerca scientifica a ricerca tecnologica.

Un gruppo di leghe metalliche che dimostravano l'effetto di memorizzazione della forma avevano ottenuto larghi consensi in molte aree di applicazione.

Questi metalli hanno un'unica struttura cristallina che cambia forma radicalmente quando soggetta ad un cambiamento di temperatura. Questo cambiamento non è continuo, come le convenzionali espansioni e contrazioni termiche, ma si ha strettamente tutto in un istante, già al di sopra di una temperatura bassa (dai 20 ai 30 gradi centigradi), che al di sotto.

Sulla base di questi materiali hanno trovato la loro via una varietà di ricerche di base sui metalli e le loro proprietà di memorizzazione di forma.

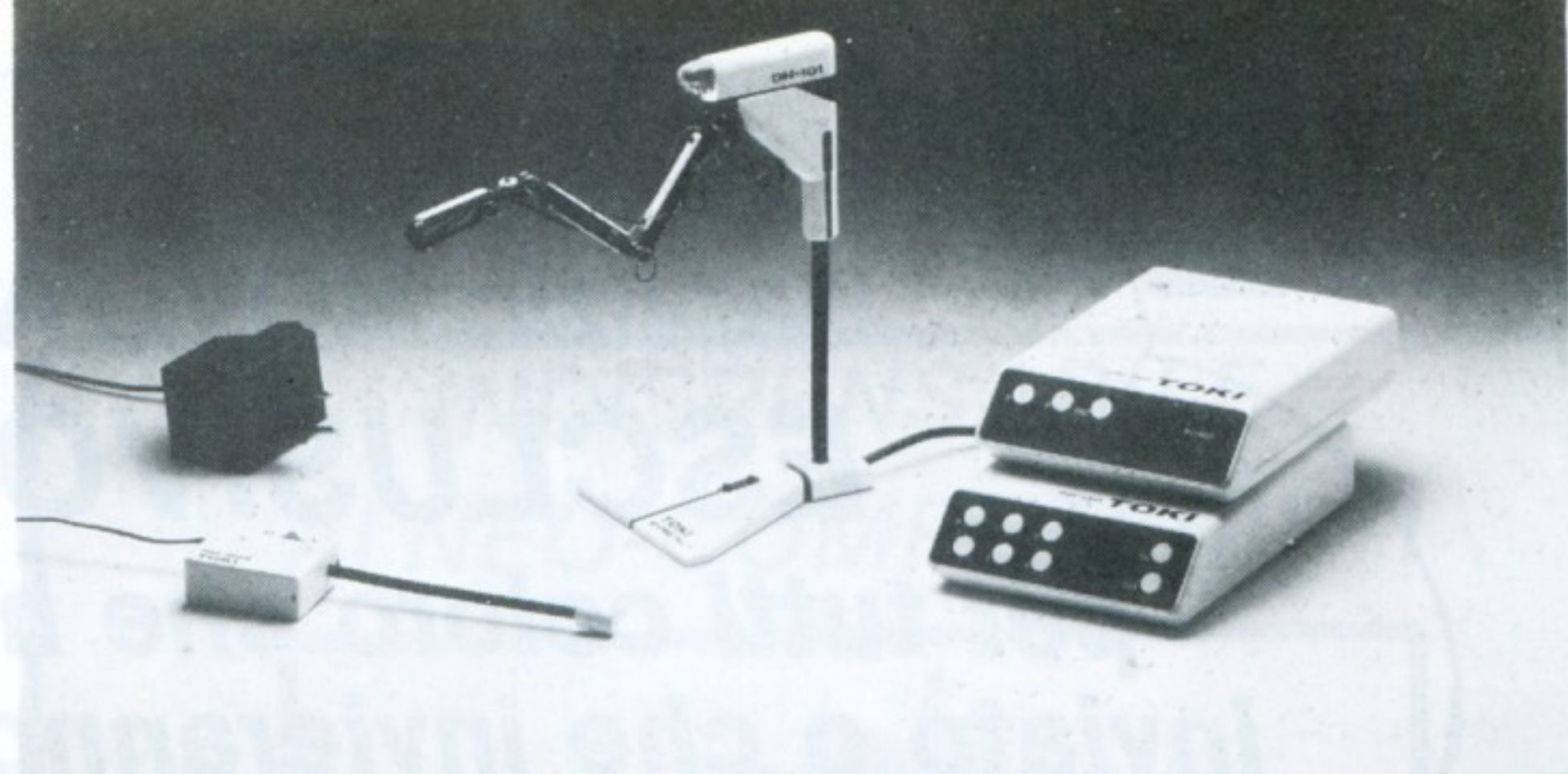
Nel 1951, due ricercatori europei, Chang e Read, dapprima notarono l'effetto SME in una lega di Oro e Cadmio e nel 1958 esposero al Brussels World's Fair un meccanismo ciclico di sollevamento di pesi.

Chang e Read mostrarono con grande chiarezza che si potevano produrre sistemi meccanici utilizzando materiali SME, perlomeno per scopi dimostrativi.

La loro ricerca si aprì verso ulteriori orizzonti e arrivarono anche a scoprire caratteristiche SME in una lega di Indio e Titanio. Comunque l'eccessivo costo di queste leghe e il loro possibile danno alla salute limitarono l'interesse nelle ricerche.

La scoperta di caratteristiche SME in una lega di Titanio e Nichel fece riconquistare interesse in questo campo di ricerca, poiché tali leghe erano considerate di costo più basso e di utilizzo più immediato.

Questa scoperta avvenne per caso alla U.S. Naval Ordnance Laboratory (ora chiamata Naval Surface Wea-



pons Center) nel 1961, da un team guidato da William Beuheler, mentre tentava di sviluppare una lega di Nichel e Titanio resistente alla corrosione e al calore. A causa del segreto militare la scoperta fu resa pubblica solo nel 1963.

Le leghe a memoria di forma TI-NI sono spesso chiamate Nitinol, «NI» e «TI» sono i simboli atomici del Nichel e del Titanio nella tabella periodica degli elementi, e «NOL» è l'abbreviazione di Naval Ordnance Laboratory.

Dopo l'annuncio il U.S. Battelle Memorial Laboratory fece vasti studi sulle proprietà del Nitinol.

Seguirono altri studi del Nitinol e possibili applicazioni creando grande interesse in numerosi altri laboratori ed università.

In Giappone lo studio delle proprietà del Nitinol e della struttura dei cristalli incominciò intorno al 1966, maggiormente presso l'università di Osaka e Tohoku. Nel 1966 l'effetto di memorizzazione della forma fu scoperto in una lega di Rame, Alluminio e Nichel. In generale tutte le quattro leghe SME possedevano le seguenti due caratteristiche:

- 1) deformazione plastica nello stato di raffreddamento o rilassamento.
- 2) rigidità crescente alle alte temperature.

Le leghe SME subiscono trasformazioni termiche di tipo elastico, cioè l'elasticità del metallo cambia a diverse temperature.

Tra il 1970 e il 1973 ricercatori trovarono la proprietà SME nelle seguenti leghe: Rame-Oro-Zinco, Rame-Titanio, Rame e Zinco, Ferro e Platino, Rame-Zinco-Alluminio, Nichel Alluminio, e Manganese Rame.

Tutti questi metalli sono in vicinanza nella tabella periodica degli elementi e hanno una struttura cristallina simile quando posti in lega.

Ulteriori studi mostrarono chiaramente che l'effetto di memorizzazione della forma era causa della fase di trasformazione tra due differenti strutture cristalline.

Tra tutte le leghe ad effetto SME studiate, la classe NI-TI di metalli mostra proprietà superiori per tensione, forza di ritorno, resistenza alla fatica, ripetibilità, non tossicità.

Oltre al Nitinol, soltanto leghe di Rame, Zinco ed Alluminio sono considerate materiali SME pratici, cioè a basso costo e di facile lavorazione.

Fabrizio Ravasi

BIOMETAL®, il risultato finale di quanto abbiamo conosciuto in questa storia, è un prodotto oggi in commercio e disponibile per qualunque tipo di ricerca e applicazione.

Questo sarà l'argomento delle prossime puntate, nelle quali mostreremo le nostre esperienze con il **BIOMETAL®** ed un simpatico robot costruito sfruttando le sue caratteristiche. Vedi IV di copertina.

NOSTRO REPORTAGE SPECIALE

smau'87

Si è tenuto a Milano dal 16 al 21 settembre lo SMAU, uno dei maggiori appuntamenti mondiali per gli operatori del settore informatico e elettronico.

Certamente LIST non poteva esimersi dall'essere presente a tale rassegna, per informare i propri lettori circa le ultime novità e tendenze del mercato informatico.

Il primo dato importante emerso dallo SMAU riguarda il ruolo del Personal Computer: esso sta cambiando.

Infatti il PC da semplice strumento di sostegno e di aiuto per pochi eletti (leggi esperti), il PC sta diventando alla portata di tutti presto farà la sua apparizione nella scuola (il Ministero della Pubblica Istruzione ha scelto lo standard MS-DOS, quello dei PC, per la scuola superiore).

Ma questo non è niente altro che un piccolo segnale.

Una recente indagine di mercato ha dimostrato che il PC si è creato un solido mercato nella fascia di utenti media e bassa, cioè in quella fascia i cui utenti sono gli hobbisti ovvero la maggior parte dei lettori di LIST. Quel settore del mercato informatico un tempo dominato dai cosiddetti "home computers" leggi C=64, Sinclair Spectrum Texas Instruments e via dicendo sta ormai diventando "re-

gno" incontrastato dei PC compatibili con costo inferiore al milione di lire. È lecito giunti a questo punto chiedersi come mai i PC compatibili stanno conquistando i nuovi utenti?

Probabilmente la risposta a questo quesito è quindi la chiave di lettura di questo "boom" dei PC si trova nel rapido e inarrestabile progresso della tecnologia informatica.

Infatti lo sviluppo della tecnologia informatica ha permesso alle grandi case costruttrici di hardware di "equipaggiare" i loro sistemi con componenti sempre più economici ma allo stesso tempo più affidabili e veloci e di aumentare considerevolmente le capacità delle varie periferiche.

Il risultato di questa "evoluzione della specie" è stato la produzione di macchine con prestazioni elevatissime e con costi ragionevolmente bassi.

A tale risultato va sommato un altro fattore importantissimo: chi compra, e quindi utilizza un PC compatibile ha a sua disposizione un parco programmi praticamente illimitato. Concludendo, cari lettori non stupitevi se nelle pagine di LIST troverete i nuovi PC della COMODORE, dell'ATARI, dell'SHR, dell'AMSTRAD e dell'OLIVETTI.

LIST si sta adeguando allo sviluppo per voi.

COMMODORE PC 1



Un Personal Computer di concezione avanzata basata su componenti a grande integrazione.

In tal modo si è potuto sviluppare, su una sola scheda madre, oltre 45 circuiti integrati che controllano tutte le funzioni, dalla memoria alla grafica video, al floppy disk controller, alle interfacce parallela, seriale e mouse.

Per questo motivo non sono necessari slots di espansione.

Il risultato è il PC 1, il personal computer più compatto, con un eccezionale rapporto prezzo prestazioni.

Insieme al sistema operativo MS-DOS 3.2 viene fornito il GW BASIC.

Prezzo Lire 995.000 IVA esclusa

SCHEDA TECNICA

Unità centrale:

Processore 8088

frequenza 4/77 MHz

coprocessore 8087 opzionale

Capacità di memoria:

512 KB RAM standard espandibile a 640 KB RAM direttamente sulla scheda madre BIOS su ROM 16 KB

Sistema operativo:

MS-DOS 3,2 GW BASIC

Memoria di Massa:

Interna: 1 floppy disk da 5", 1/4. 360 KB formattati

Esterna:

1 floppy disk da 3" 1/2 o
1 floppy disk da 5" 1/4

Interfacce:

Parallela Centronics 8 bit
Seriale RS 232 C
RGB 1 video composito b/n
Porta mouse compatibile
Microsoft

Video Display:

MDA
CGA
HERCULES GRAPHICS 720x348
pixel
Testo monocromatico 80x25

Video OUTPUT:

RGB I digitale
Monocromatico
Composito b/n

Modo Grafica:

320x200 4 colori
640x200 2 colori

MODO TESTO:

40x25 16 colori
80x25 16 colori

Porte di Espansione:

1 slot esterno

Tastiera:

84 tasti, separata, compatibile PC 40

Monitor:

Monocromatico, colore opzionale

Dispositivi Esterni del Sistema:

Bus per espansione esterna

Compatibilità:

Totalmente compatibile con
PC 10/20/40 MS-DOS

Dimensione CPU:

33x32x8,5 cm

È un vero home professional. Potente
compatibile, completo compatto.

Caratteristiche: Sistema operativo MS-
DOS 3.2, microprocessore NEC V 40
compatibile, alta velocità (8 MHz), 512
Kb espandibile 640 Kb, disk drive 3.5"
Opzioni: secondo floppy disk 3.5",
stampante DM 91, Monitor a colori. Gra-
phics controller hard disk unit controller,
TV/Telematic adapter, CDROM, modem,
adattatore Pagine Gialle elettroniche/VIDEOTEL, Music Show.

Software: 100 titoli originali Libreria Oli-
vetti Prodest, oltre alle migliaia di pro-
grammi MS DOS esistenti sul mercato.

Prezzo configurazione standard 1.195.000
IVA esclusa.

SCHEDA TECNICA PC 1
Hardware
Processore:

NEC V40 — 8088 compatibile —
4.77/8 Megahertz

RAM utente:

512 Kb Espandibile via bus connec-
tion a 640 Kb

ROM:

32 Kb per diagnostica e Bios

Tastiera:

83 tasti con sezione alfanumerica,
10 funzioni e pad numerico

Monitor:

Modi grafici (CGA) — 640x200
B/W, 320x200 4 colori, 160x200 16
colori

Alfanumerico (font 8x8) — 80x25:
40x25

Floppy disk:

3.5" doppia faccia/doppia densità
Capacità: 720 Kb

Joystick/ Mouse:

In emulazione cursori

Interfacce:

Parallele tipo Centronics, seriale-
RS232, suono (HI/FI), floppy
disk (FDU esterno da 5,25"), ex-
pansion bus (IBM compatibile)

Opzioni:

Via expansion bus è possibile con-
nettere un box per due card,
small/half size

Il box potrà ospitare opzioni quali:
RAM, LAN, Modem, Graphics
controller, CD ROM controller,
TV/Telematic adapter, Hard disk
unit controller, Music Show, adat-
tatore Pagine Gialle elettroniche/VEDEOTEL.

PC ATARI

Sistema Operativo MS-DOS

Con questo personal computer, Atari of-
fre una reale alternativa a tutti gli acqui-
renti che in seguito desiderano utilizzare
software per il sistema operativo MS-
DOS.

È infatti compatibile con le schede grafi-
che CGA, EGA ed Hercules.

Il clock del processore a 4,77 MHz può
essere portato a 8 MHz.

Con i suoi 512 Kb RAM, il PC Atari si
presenta con una memoria ampiamente
sufficiente.

Tutte le periferiche della serie di computer
ST possono essere collegate al PC. So-
prattutto per quanto riguarda gli ottimi
floppy della serie ST Atari a 3 1/2 pollici.
Il disco rigido può infatti essere collegato
direttamente.

Il PC Atari viene offerto, con monitor
PCM 124, a Lire 990.000 (IVA esclusa).

OLIVETTI PRODEST PC1

**MULTITECH
PC ACER - 500 +**


Descrizione:

8088-4,77/8 MHz, 256KB RAM
esp. 640KB su scheda base
4 slot di cui 2 liberi
1 Floppy D. 5,25"
IBM XT comp., tastiera scheda video
MGA (monocrom. Grafica)
Monitor monocrom. 12"
1 porta seriale, 1 porta parallela,
MS/DOS
DIM. 400x120x380mm
Prezzo Lire 990.000 IVA esclusa

AMSTRAD PC-1512



Descrizione

- Unità di sistema a 512K con tutta l'elettronica su un'unica scheda.
- Processore 8086 a 8 MHz.
- Memoria espandibile fino a 640K su scheda principale.
- Adattatore grafico speciale con modo in alta risoluzione a 16 colori o 16 toni di grigio.
- Tre slot di espansione per schede a tutta lunghezza con presa di alimentazione disponibile nell'unità di sistema per disco fisso e controller.
- Scelta fra uno o due dischi da 5" e 1/4 oppure una unità dischi e un disco fisso da 20 megabyte.
- Le versioni con disco fisso sono dotate di manuale e dischetto supplementare con le unità di backup, restore e format.
- La scheda principale comprende un'interfaccia seriale RS232C e una parallela Centronics con connettori standard.
- Altoparlante con regolatore di volume.
- Orologio in tempo reale e configurazione RAM alimentate da batterie.
- Zoccolo per processore matematico 8087.
- Connettore per penna ottica o Joystick.
- Completo di monitor a colori o monocromatico (16 livelli di grigio).
- L'adattatore grafico standard incorporato comprende le seguenti modalità:
Media risoluzione Alfa - 16 colori, 40x25 caratteri; Media risoluzione Grafi-

ca - tre tavolozze a 4 colori 320x200 punti;
Alta risoluzione Grafica - 2 colori, 640x200 punti;

+ Grafica speciale HiRes - 16 colori, 640x200 punti.

Questo modo grafico speciale è di normale dotazione in aggiunta ai modi Alfa e Grafici e consente di visualizzare della grafica in alta risoluzione a 16 colori (per esempio GEM).

- Tastiera QWERTY italiana completa.
- Tasti di Blocca Numeri e Blocca Maiuscole con spia luminosa.

Porta per joystick o penna ottica sulla tastiera.

Tasti aggiuntivi di Can e IMMISSIONE, Piedini regolabili.

- Mouse e due pulsanti con Porta dedicata nell'unità di sistema.

Comando MOUSE. COM compatibile Microsoft con altre speciali funzioni per testi.

- Software operativo esteso incluso in 4 dischetti (tutto in italiano).

Compatibile BIOS ROM.

Sistema Operativo MS-DOS 3.2

Device driver compatibili con disco RAM ed estesa flessibilità per i supporti.

Ampia dotazione di utilità compatibili.

Ammette l'uso di dischi fissi e reti locali.

GEM (Graphics Environment Manager) + GEM Desktop v. 2 Digital Research.

GEM Paint Digital research.

Basi 2 interprete BASIC della Locomotive Software funzionante in GEM.

Sistema Operativo DOS Plus Digital Research che consente l'esecuzione di applicazioni MS-DOS e CP/M-86.

- Manuale d'uso in italiano.

Comprende istruzioni per l'avvio del sistema, dettagli sull'uso del GEM DOS, delle utilità, del DOS Plus e un sommario del BASIC 2 Locomotive.

IBM PC-SYSTEMS/2



SCHEMA TECNICA dei PC-Systems/2

Modello 30-021

Microprocessore	8086
Velocità microprocessore	8 MHz
Memoria RAM	640 Kbytes
Sistema operativo	DOS 3.3
Floppy Disk	3.5 pollici
Capacità floppy disk	720 Kb
Capacità hard disk	20 Megabytes
Costo	2295 dollari

Modello 50-021

Microprocessore	80286
Velocità Microprocessore	10 MHz
Memoria RAM	1 Megabyte
Sistema operativo	DOS 3.3-OS/2
Floppy disk	3.5 pollici
Capac. Floppy disk	1.44 Megabytes
Capacità hard disk	44 Megabytes
Costo	5295 dollari

Modello 80-041

Microprocessore	80386
Velocità Microprocessore	16 MHz
Memoria RAM	1 Megabyte
Sistema operativo	DOS 3.3-OS/2
Floppy disk	3.5 pollici
Capac. Floppy disk	1.44 Megabytes
Capacità hard disk	44 Megabytes
Costo	6996 dollari

Modello 80-111

Microprocessore	80386
Velocità Microprocessore	20 MHz
Memoria RAM	2 Megabyte
Sistema operativo	DOS 3-OS/2
Floppy disk	3.5 pollici
Capac. Floppy disk	1.44 Megabytes
Capacità hard disk	115 Megabytes
Costo	10995 dollari

Macro Assembler per Atari ST e Amiga

La Metacomeo, società «autrice» dell'Amiga DOS ha presentato un pacchetto software di Macro Assembler per l'Amiga, denominato Termed MCC Assembler.

Questo pacchetto software, che è inoltre disponibile per l'Atari ST, basato sul microprocessore 68000 della Motorola è composto da un Linker e da una libreria di programmi per lo screen editor.

I programmi del Motorola 68000 potranno così scrivere in maniera rapida e sollecita efficienti routines usando le macro

istruzioni che sono totalmente compatibili con il set e le specifiche delle istruzioni del Motorola.

L'intero pacchetto applicativo è fornito assieme ad un floppy disk introduttivo, un manuale d'uso e un manuale di installazione.

Il prezzo di vendita di questo pacchetto è di 69.95 dollari.

PHILIPS

NMS 8245: il computer facile



Quando si pensa ad un computer si è indotti a credere che un sistema valido sia necessariamente costoso.

Questo fino a ieri.

Da oggi Philips mette a disposizione del pubblico un sistema che risponde a quattro requisiti fondamentali. Estrema semplicità di impiego, completezza, flessibilità ed un prezzo competitivo fanno infatti di **NMS 8245** uno dei computer più appetibili per professionisti, commercianti e studenti che troveranno nel nuovo pacchetto Philips la soluzione più attuale per le più diverse applicazioni.

NMS 8245 Philips si avvale di un pacchetto applicativo integrato denominato EASE; iniziale di Enhanced Application Software Environment, ma, per tutti gli utilizzatori EASE, significa semplicemente «**FACILE**».

Finalità principale del pacchetto è infatti quella di permettere a chiunque di avvicinarsi alle applicazioni del computer, anche senza alcuna conoscenza di base.

Le applicazioni dei testi EASE sono:

- L'elaborazione dei testi (Word Pro).
- La piccola editoria personale (Desk Top Publisher).
- La gestione dei dati (Database).
- Il foglio di calcolo elettronico (Calcform).
- La creazione di grafici finanziari e gestionali (Charts).

Inoltre in ogni momento è possibile richiamare sul video una calcolatrice, un notes per gli appunti oppure l'agenda appuntamenti.

Il programma è gestito a finestre con menu a simboli (icone). L'utilizzo della maggior parte delle opzioni avviene ancor più facilmente mediante un mouse (**SBC 3810**).

Ogni singolo modulo del pacchetto applicativo EASE può essere facilmente collocato e integrato con gli altri, per fare ad esempio mailing personalizzate, per certificare i dati di una tabella, per integrare a video o in un'unica pagina stampata testi, tabelle e grafici con layout studiati ad hoc.

L'NMS 8245 è dotato di memoria ROM da 64 Kbytes, memoria RAM da 256 Kbytes, tastiera integrata da 73 tasti, floppy disk drive incorporato da 3,5".

Dispone inoltre di interfaccia per monitor RGB, CVBS o radiofrequenza per il collegamento alla presa di antenna di qualunque televisore domestico.

Oltre al software EASE a corredo è presente il sistema operativo MSX-Dos e il programma Designer Plus, con il quale è possibile fare qualunque tipo di disegno.

Il prezzo al pubblico di **NMS 8245** è di sole 800.000 lire iva inclusa, un prezzo che ne fa senza dubbio uno dei pacchetti più convenienti attualmente sul mercato. Con **NMS 8245**, Philips riconferma la sua precisa caratteristica di Azienda specializzata in computer «a misura d'uomo», una specializzazione che Philips affronta con proposte ampie, diversificate e fortemente interessanti dal punto di vista prezzo.

PHILIPS

La casa Telematica comincia con NMS 8280



Il 1986 è stato per il personal computer un anno positivo.

Il mercato ha registrato un netto incremento in volume.

Il computer che non è più solo "personal" né solo "home" diventa con la tecnologia Philips il cuore dei sistemi elettronici di casa.

Philips NMS 8280 è quindi il vero primo personal computer per la casa che consente anche il collegamento con banche dati, (videotel, pagine gialle...), diventa un sofisticato strumento musicale sia per il principiante che per il più esigente professionista.

Uno strumento quindi non più solo per lavorare, ma anche per lo studio, lo svago, gli hobby, di quotidiana utilità anche tra le pareti domestiche.

NMS 8280 consente inoltre:

Il collegamento con le banche dati. Grazie al "Modem NMS 1255" e al programma dato a corredo è possibile infatti inviare programmi creativi Immagini digitalizzate ed accedere a banche dati per usufruire di migliaia di notizie e informazioni.

La musica computerizzata (programma NMS 1205)

Anche chi non sa nulla di musica può divertirsi suonando con gli accorgimenti elaborati dal computer.

Con l'aggiunta della tastiera NMS 1160 il sistema musicale Philips diventa professionale offrendo anche a un musicista provetto la possibilità di comporre della splendida musica.

NMS 8280 Philips sarà disponibile ad un prezzo estremamente interessante (circa Lire 1.700.000) uno stimolo ulteriore per avvicinare la famiglia al mondo magico dei computers.

SCHEDA TECNICA NMS 8280

Microprocessore:

Z80

Memoria:

ROM 64KB; RAM utente 128KB; VIDEO RAM 128KB
unzione di ramdisk/memory mapping

Risoluzione in modo testo:

40/80 colonne - 24 righe (modi)

Risoluzione in modo grafico:

512-212 punti (7 modi)

Numero colori:

256 selezionabili su una tavolozza di 512

Numero Sprite/Linea:

8 a 16 colori diversi per linea

Tastiera:

Professionale a 73 tasti completa di tastierino numerico separato a 16 tasti

Interfacce:

Televisione canale 36
Monitor CVBS
Monitor RGB Scart
2 SLOT per cartucce
Connettore Audio/Video A
Connettore Audio/Video B
Connettore Audio/Video In/Out
2 prese per Joystick
Mouse/Tavoletta grafica
Presa registratore
Presa stampante Centronics

Floppy disk drive:

Due unità incorporate da 720KB formattati ciascuna

Software incluso:

Elaborazione grafica Home Office-2:

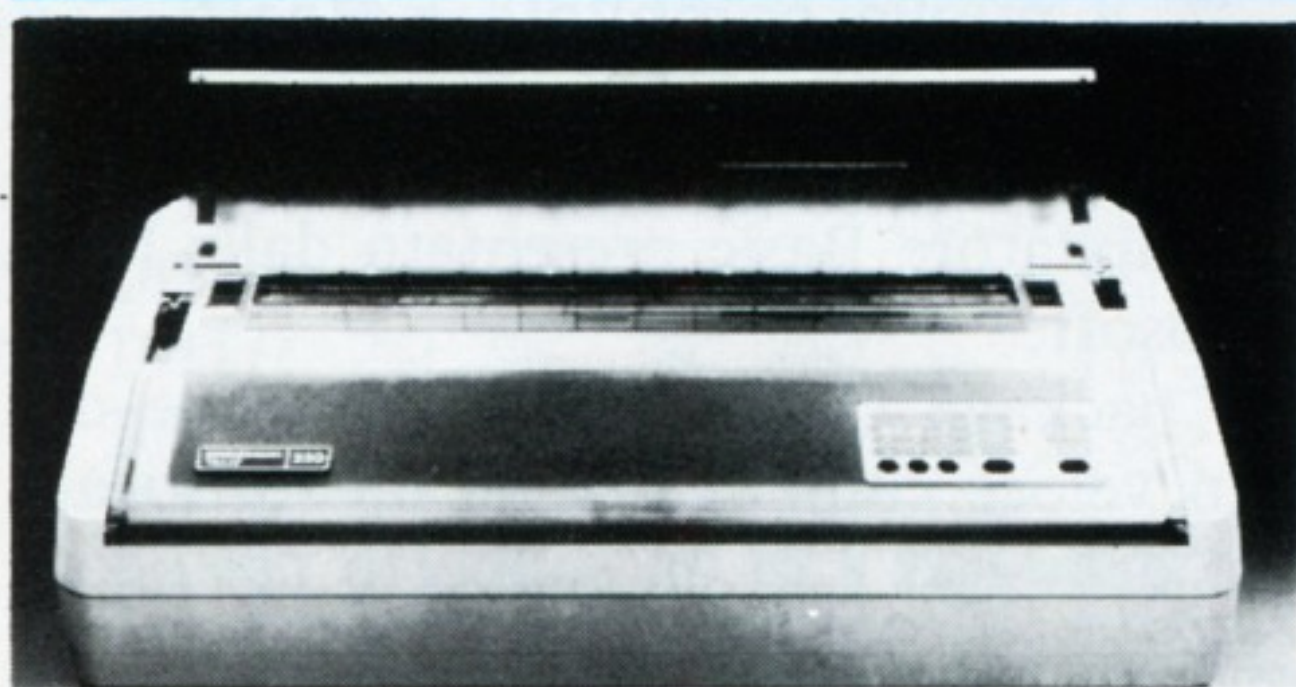
- Elaborazione testi
 - Gestione archivi
 - Tabellone elettronico
 - Grafica finanziaria
 - Agenda appuntamenti
 - Pianificatore di risorse
- MSX-DOS sistema operativo

A corredo:

Mouse
Cavo antenna
2 cavi video per collegare Videoregistratori
Camera recorder
Tv/monitor

MANNESMANN TALLY

Novità Smau 1987



Ideali per WP e EDP le nuove MT 230 a 9 o 18 AGHI a 300 CPS

Proseguendo nell'ampliamento della propria gamma di stampanti studiate su misura per le più diverse esigenze e le più svariate applicazioni, la Mannesmann Tally presenta in occasione del XXVI Smau la nuova serie di stampanti MT 230: capaci di 300 cps in draft e di 150 in Alta Qualità.

Le stampanti della serie MT 230 sono disponibili in due versioni, che differiscono in alcuni particolari come la testina di stampa, che le specializza nel loro campo di impiego. Il modello MT 230/9 ha 9 aghi ed è indirizzato ad impieghi EDP, mentre il modello MT 230/18 ha 18 aghi ed è ideale per WP e OA.

Come tutti i prodotti MANNESMANN TALLY, anche le stampanti della nuova serie sono state progettate e prodotte per sopportare elevati carichi di lavoro sia in modo "draft" che in alta definizione. Quest'ultima modalità è particolarmente efficace nel trattamento testi e permette di ottenere stampe di qualità senza sacrificare nulla alla velocità operativa e alla versatilità.

Alle caratteristiche di robustezza e silenziosità, le nuove stampanti MT 230 aggiungono una particolare duttilità nella gestione della carta: il caricamento del modulo continuo infatti avviene con un trattore a trascinamento o con un trattore di spinta; quest'ultima permette lo strappo facilitato della carta. L'alimentazione a foglio singolo invece avviene frontalmente per mezzo di un rullo a frizione che serve anche per il caricatore automatico opzionale.

L'elettronica dei modelli MT 230 è fra le più sofisticate e consente di personalizzare a seconda delle proprie esigenze l'interfaccia di collegamento con l'elaboratore e le modalità di emulazione. Sono disponibili "Smart Card" opzionali che consentono di gestire il colloquio con il computer in altre emulazioni oltre a ANSI, IBM-PC e EPSON FX105, già presenti nel firmware della stampante.

ATARI

Stampante Laser SLM 804

Caratteristiche:

- 8 pagine al minuto.
- Risoluzione, 300 punti per pollice.
- Formato del foglio, A4 standard (216x297 mm).
- Alimentazione dei fogli automatica e manuale.
- capacità della cassetta di alimentazione 150 fogli.
- Utilizza l'interfaccia ATARI ST DMA.
- Emulazione standard di DIABLO 630 e GDOS.

La stampante Laser Atari SLM presenta l'indubbio vantaggio di offrire la potenza della tecnologia laser agli utenti dell'ATARI ST.



Il risultato è una stampa di alta qualità con una risoluzione di 300 punti per pollice. In 1 mm², ad esempio sono contenuti 140 punti.

La SLM può stampare otto pagine al minuto, a una velocità quindi decisamente superiore rispetto a una stampante marginale di qualità comparabile.

Non solo. Con l'Atari SLM si possono stampare anche immagini grafiche.

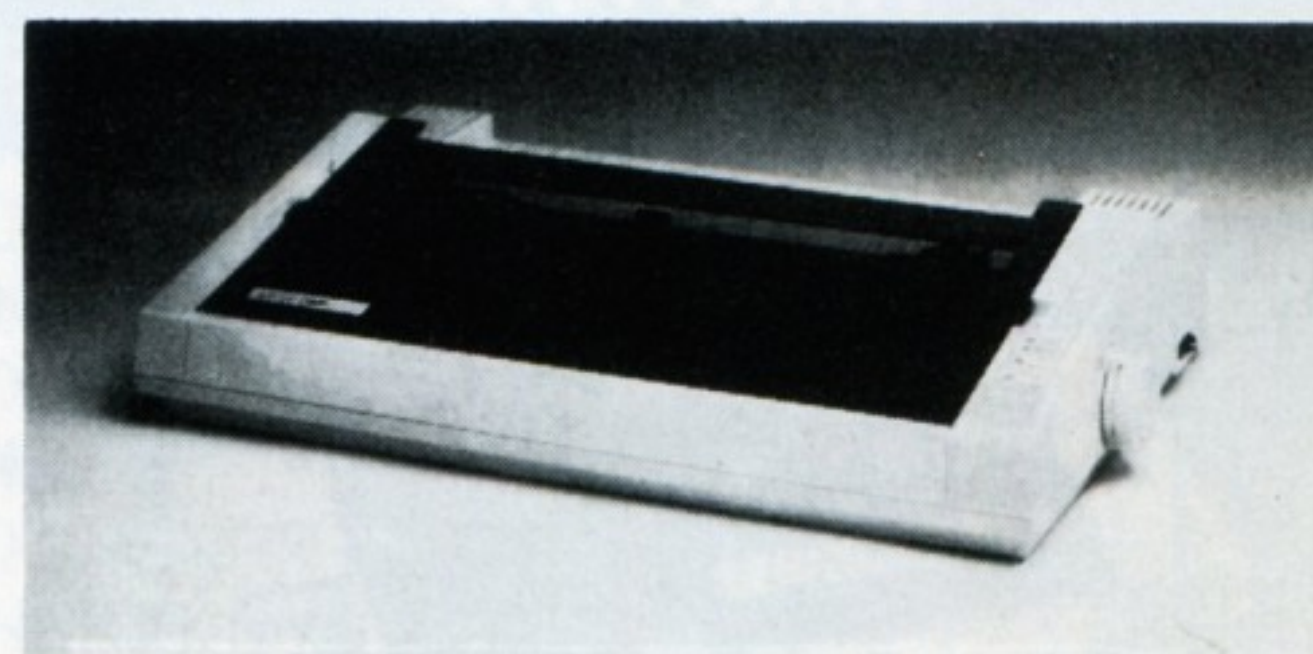
È cioè possibile stampare qualsiasi immagine o grafico utilizzando svariati font di caratteri, sempre con un alta risoluzione. La SLM si affida alla grande capacità di memoria e alle prestazioni del veloce processore del Mega ST.

La costruzione della stampante Laser è semplificata perché gran parte dell'intelligenza e delle risorse computazionali richieste sono fornite dal ST.

Il trasferimento rapido dei dati tra il ST e la stampante Laser viene effettuato tramite la porta DMA. Questo determina alte prestazioni nella stampa di fogli singoli. Anche per questo prodotto ATARI si è preoccupata di mantenere un prezzo contenuto: Lire 2.790.000 (IVA esclusa) con incluso un programma di editoria elettronica.

AMSTRAD

Stampante DMP 4000 a 132 Colonne



La AMSTRAD introduce la nuova stampante DMP 4000 a matrice di punti con rullo da 15 pollici per 132 colonne di stampa a 10 caratteri per pollice e con velocità di 200 caratteri al secondo.

La stampante DMP 4000 funziona con l'intera gamma di personal computer della AMSTRAD e con qualsiasi altro personal e home computer dotato di interfaccia



standard parallela e costituisce la stampante disponibile oggi con le caratteristiche più avanzate.

Più di cento diverse combinazioni di stili di caratteri, inclusa la stampa "near letter quality" (NLQ) unitamente all'insieme completo di caratteri ASCII ed all'insieme di caratteri grafici IBM, costituiscono la soluzione ad ogni problema di stampa.

In aggiunta, le possibilità di indirizzamento del singolo punto permette la stampa di ogni tipo di grafico e la completa compatibilità con lo standard EPSON permette alla stampante DMP 4000 di operare correttamente con i più diffusi packages di software applicativo.

Il prezzo al pubblico di questa stampante è di lire 649.000 (IVA esclusa).

SCHEDA TECNICA

Sistema di Stampa:

Matrice di punti ad impatto

Velocità di stampa:

200 CPS (caratteri standard) 50 CPS (caratteri NLQ)

Caratteri:

96 Ascii
96 corsivo
96 NLQ
96 NLQ corsivo
9 set caratteri internazionali
132 caratteri speciali

Numero Colonne:

136 standard
233 condensato

Velocità di Interlinea:

100 m/sec

Interfaccia:

Parallela (compatibile Centronics)

AMSTRAD

Stampante DMP 3160 a 160 CAR/SEC



La Amstrad amplia la sua gamma di stampanti introducendo la nuova DMP 3160 a matrice di punti con stampa su 80 colonne a 10 cpi (caratteri per pollice) e velocità di 160 caratteri al secondo.

La stampante DMP 3160 funziona con

l'intera gamma di personal computer prodotti dalla AMSTRAD e con qualsiasi altro personal computer dotato di interfaccia standard parallela.

La DMP 3160 accetta tutti i codici di controllo dello standard EPSON e supporta l'insieme completo di caratteri grafici IBM oltre a permettere più di cento diversi stili di caratteri compresa la stampa di testi in forma compatta, espansa e "Near Letter Quality" (NLQ) e la stampa di grafici.

Fogli singoli o moduli continui possono essere alternativamente usati ad uno speciale sistema "flatbed" permette una facile introduzione frontale della carta ed il suo allineamento sia nel funzionamento a frizione che quello con trattori.

Accanto alla nuova DMP 3160 la AMSTRAD offre la stampante DMP 3000 con le medesime caratteristiche della precedente ma con velocità di stampa ridotta a 100 caratteri al secondo.

Il prezzo al pubblico della DMP 3160 è di Lire 369.000 mentre il modello DMP 3000 viene 349.000.

Naturalmente i prezzi si intendono IVA esclusa.

SCHEDA TECNICA

Sistema di Stampa: matrice di punti ad impatto

Velocità di Stampa:

160 CPS (caratteri standard)
40 CPS (caratteri NLQ)

Dimensione Carattere:

2.1 x 2.55 millimetri

Dimensioni di Stampa:

Standard 10 CPI/80 CPL
Mini 12 CPI/96 CPL
Condensato 17 CPI/137 CPL
Doppia larghezza standard 5 CPI/40 CPL
Doppia larghezza Mini 6 CPI/48 CPL
Doppia larghezza condensato 8.5 CPL/68

Numero Colonne

80 standard
40 doppia larghezza
132 condensato
66 doppia larghezza condensato

Interfaccia:

Parallela (Centronics compatibile)

Alimentazione:

220/240 volt c.a. a 50 Hz

Dimensioni:

400 x 250 x 100 millimetri

Peso:

4.2 Kg

COMMODORE

DISK DRIVE 1581



L'introduzione di questo nuovo disk 1581 da 3,5" mette a disposizione dei modelli C-64 e C-128 il meglio della tecnologia sviluppata sui dischi.

I principali vantaggi di questa tecnologia sono costituiti da una maggior velocità nel trasferimento dei dati, una maggior capacità di memoria per l'archiviazione ad una miglior protezione da incidenti fortuiti che possono occorrere al disco stesso. Rispetto ai disk da 5"1/4, i nuovi da 3,5" sono più compatti e hanno una solida protezione per prevenire eventuali deformazioni o abrasioni dei supporti magnetici.

La capacità del modello 1581 è di 880 KB, ideale anche per gli utilizzatori più esigenti.

L'interazione tra programmi e disco è più veloce grazie alla velocità di trasferimento dei dati che, nel modello 1581, è tre volte superiore rispetto a quella degli attuali disk drive del C-64.

Il nuovo 1581 è totalmente compatibile con i modelli C-128, C-64, PLUS4, e C-16.

MICROSOFT

Presenta:

il nuovo compilatore Basic

Si chiama QuickBasic 3.0 il nuovo compilatore Basic presentato dalla Microsoft.

Il package QuickBasic si compone di un veloce compilatore, di un sofisticato debugger e di un supporto per i co-processori matematici 8027 e 80287.

Il programma di debugger è basato sul «Code View Debugger» fornito assieme al compilatore C della Microsoft stessa.

Esso inizia ad operare durante l'esecuzione di un programma mediante la pressione dei tasti control-break.

È possibile dunque, grazie a questo potentissimo debugger determinare gli errori seguendo passo dopo passo l'esecuzione del codice sorgente.

COMMODORE C 64 IN CONFEZIONE TELEMATICA.



È possibile inoltre, esaminare il contenuto delle variabili mentre il programma è in fase di esecuzione e inserire dei «break-points» per arrestare l'esecuzione del programma.

Un'altra caratteristica interessante di questo compilatore Basic è quella che permette all'utente di dividere lo schermo in finestre e quindi «vedere» contemporaneamente il codice sorgente, i valori delle variabili e l'output del programma, il tutto mentre il programma stesso è ancora in fase di esecuzione. Il supporto per il coprocessore matematico permette, nel caso che l'utente disponga del coprocessore stesso, di generare un codice molto veloce mentre qualora l'utente non disponesse di tale coprocessore il supporto «setta» automaticamente le varie routine software che permettono di emulare il funzionamento del coprocessore matematico.

Vi sono inoltre a disposizione dell'utente numerose altre caratteristiche inerenti la programmazione come ad esempio il supporto per le variabili locali e globali, la possibilità di utilizzare labels alfanumeriche per le linee al posto dei «classici» numeri di linea e la compilazione separata che permette la divisione dei programmi in moduli e la possibilità di compilare ogni singolo modulo in maniera indipendente. Inoltre è possibile riunire i singoli moduli in una «libreria» e utilizzarli in altri programmi senza bisogno di doverli ricompilare.

La Microsoft ha anche aggiunto una tecnica di ricerca ed individuazione degli errori molto sofisticata.

Infatti, alla fine di ogni compilazione, sarà il compilatore stesso a posizionare il cursore sull'errore individuato e che dovrà essere corretto.

Il compilatore QuickBasic 3.0 offre la totale compatibilità con lo standard del linguaggio Basic e supporta il modo grafico esteso della EGA.

Per poter utilizzare il QuickBasic 3.0 occorre un PC I.B.M. o un suo compatibile con almeno una memoria RAM di 320 Kbytes.

Inoltre è richiesta una release del sistema operativo MS-DOS superiore almeno alla versione 2.0.

C'è da osservare che il QuickBasic supporta sia floppy disk da 5" 1/4 che 3" 1/2.

Il costo di questo pacchetto software è di 99 dollari U.S.A.

Per ulteriori informazioni:

**Microsoft S.p.A. - Via Michelangelo, 1
20093 Cologno Monzese (MI) -
Tel.: 02/2549741**



Da oggi il tuo tempo libero acquista una nuova dimensione con COMMODORE, SEAT e SIP.

Prova ad immaginare quante altre cose belle puoi fare oltre ai giochi ed ai programmi, trasformando il C-64 in un terminale telematico.

Oltre al Computer, infatti, la scatola contiene una grande novità completamente gratuita: l'**Adattatore Telematico**. Questo ti permette di oltrepassare i confini della tua stanza e di aprire una finestra sul mondo.

L'adattatore telematico ti consentirà infatti di entrare in contatto con servizi di estremo interesse per un appassionato di computers come te.

VIDEOTEL, un servizio con il quale potrai collegarti 24 ore su 24 con altre 250 Banche Dati e ricevere videogiochi o programmi, informazioni e servizi inerenti il tempo libero, il turi-

smo, la scuola e utilizzare un servizio di casella postale per lo scambio di messaggi con tutti gli altri abbonati.

PAGINE GIALLE ELETTRONICHE, un servizio che sviluppa e diffonde in tempo reale informazioni di carattere anagrafico e tecnico-commerciale. Questo collegamento può portare anche i tuoi genitori ad una conoscenza più approfondita del mondo delle aziende.

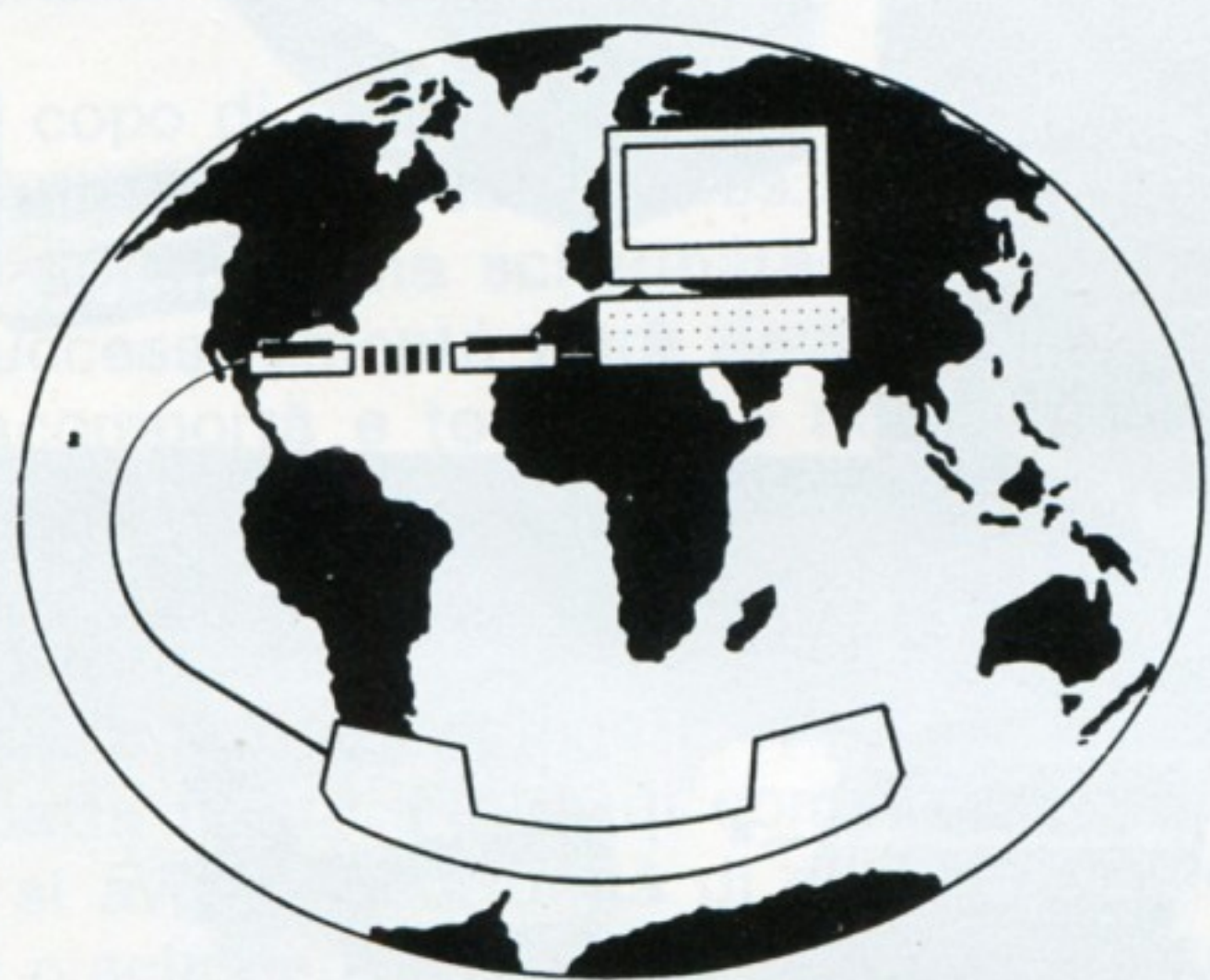
La scatola telematica 1 contiene:

- 1 C-64 C
- 1 1530 Registratore di dati
- 1 Geos sistema operativo (da usare con drive 1541)
- 1 6499 Adattatore Telematico

La scatola telematica 2 contiene:

- 1 1541 Unità a dischi
- 1 1351 Mouse
- 1 6499 Adattatore Telematico

Nascita e sviluppi della TRASMISSIONE



Il Modem, questo

Eccoci al nostro secondo appuntamento; proseguiamo il discorso sulla trasmissione dati lasciato in sospeso (sul più bello!) lo scorso numero.

Come abbiamo già detto, le tecniche di trasmissione dati si sono andate sviluppando per far fronte all'esigenza di scambio di informazioni tra elaboratori o tra terminali periferici ed elaboratore. Con il termine «periferici» intendiamo anche quelli geograficamente distribuiti.

Per garantire una connessione capillare, quale soluzione migliore se non quella di utilizzare la rete telefonica?

In questo modo si ottenne un duplice vantaggio: le connessioni erano già esistenti, in quanto la rete dati veniva a coincidere con la rete telefonica mondiale; un'elevata standardizzazione, che garantiva la più ampia possibilità d'interconnessione tra le reti telefoniche di paesi diversi.

Sorgeva ora un problema: la rete telefonica, progettata ed utilizzata per il trasporto di segnali analogici, propri delle trasmissioni telefoniche, risultava inadatta al trasporto dei segnali digitali degli elaboratori. Occorreva quindi che, prima dell'inoltro, tali segnali venissero amplificati per ovviare alle forti dispersioni sulla linea, resi compatibili con gli standard telefonici, e che al ricevimento fosse compiuta l'operazione inversa.

Queste due operazioni vennero affidate ai MoDem, abbreviazione di MODulatore/DEModulatore.

Il loro compito è quello di modulare il segnale in partenza, e di demodulare quello in arrivo (vedi figure), facendo variare uno dei tre parametri caratteristici della portante: frequenza, fase, ampiezza; da cui poi prendono nome le tre tecniche di trasmissione: PSK (Phase Shifting Key), FSK (Frequency Shifting Key) e QAM (Quadrature Amplitude Modulation).

Nella letteratura specializzata, il Modem viene spesso indicato anche con il termine DCE (Data Circuit terminating Equipment), mentre i calcolatori e i terminali vengono indicati con DTE (Data Terminal Equipment).

Esistono due tipi di Modem. In «banda fonica», di cui abbiamo appena parlato, che utilizzano la normale linea telefonica; in «banda base», utilizzati quando la trasmissione avviene su linee appositamente studiate ed installate per la trasmissione dati, quindi con elevata qualità del segnale e costituite da speciali cavi che offrono elevatissime velocità di trasferimento e ampia banda di trasmissione, che consente di trasmettere i segnali così come partono dal terminale, senza doverli amplificare ma semplice-

■ Ministoria
dei Modem

■ Modem
in banda base e
in banda fonica

■ Trasmissioni
sincrone
e asincrone

■ Standard Hayes

NE DATI

di Fabio Scafati

sconosciuto

mente adattandoli elettricamente alla rete telefonica. Essendo quindi più semplici dei Modem in banda fonica, il costo dei Modem in banda base è inferiore, il che ne favorisce l'uso e la diffusione quando possibile.

Le modalità di trasmissione oggi utilizzate sono due: asincrona (detta anche start/stop) e sincrona. La prima, prende origine dalle comunicazioni telescrivente-elaboratore, in cui la velocità di battitura dell'utente sulla tastiera è contenuta e non costante; quindi la successione temporale degli istanti in cui vengono inviati i caratteri è, di fatto, casuale: può esistere un intervallo di tempo qualunque tra la fine di un carattere e l'inizio del successivo. Ogni carattere inviato, è preceduto allora da un segnale di inizio (bit di START) e seguito da un segnale di fine (bit di STOP). L'apparato ricevente è così in grado di identificare il carattere trasmesso e di prelevare dalla linea. Questa modalità, peraltro molto utilizzata, è relativamente inefficiente; infatti per ogni carattere (8 bit) servono due bit di controllo (bit di START e di STOP). Quindi per trasmettere 8 bit di dati, occorre inviarne 10, con un sovraccarico del 20%.

La seconda, molto più efficiente alle elevate velocità, è utilizzata dall'utenza professionale. Viene detta «sincrona» in quanto i caratteri non vengono più trasmessi uno alla volta, ma in «pacchetti» di decine o centinaia di caratteri. Ogni pacchetto inviato, è preceduto da un segnale di sincronismo (SYN) originato dal Modem che trasmette. Grazie a tale segnale, il Modem ricevente sa in quali istanti prelevare i caratteri del pacchetto, dalla linea. La fine del pacchetto, è segnalata anche qui da un carattere di chiusura e da uno o più caratteri di controllo, atti a rilevare eventuali errori di trasmissione. In modalità sincrona, vengono utilizzati solo pochi caratteri in più oltre quelli del messaggio, e il sovraccarico è trascurabile.

Vediamo ora come si presenta fisicamente un Modem comunemente utilizzato per trasmissioni non professionali (asincrone). Ha la forma di un contenitore grande circa come una scatola di floppy disk, generalmente in alluminio o lamiera verniciata. Frontalmente presenta numerose spie, utili nel corso della trasmissione, e a volte, uno o più commutatori. Sul retro sono presenti l'uscita RS-232, che rappresenta lo standard di comunicazione asincrona nel mondo PC ed Home; l'alimentazione, uno o due collegamenti alla rete telefonica, l'interruttore di accensione e, a volte, il controllo del volume dell'altoparlante interno (se c'è).

I led presenti sul frontalino aiutano lo «smanettone» a controllare la trasmissione. Quello sempre presente su qualunque tipo di Modem, è il CD (Carrier Detect), rivelatore della portante telefonica, ovvero di avvenuto «aggancio» del Modem situato sull'altro capo della linea. Nei Modem più «lussuosi» si possono trovare altre spie: MR (Modem Ready), TR (Terminal Ready), SD (Send Data), RD (Receive Data), HS (High Speed), AA (Auto Answer), OH (Off Hook).

Quando si accende il Modem, si illumina la spia MR: Modem pronto; l'accensione del led TR indica la disponibilità, del terminale o calcolatore, a ricevere o inviare comandi; SD lampeggia quando vengono inviati dei dati, RD quando vengono ricevuti; HS si illumina quando il collegamento è ad alta velocità (1200 o 2400 baud); AA è attiva quando il Modem è abilitato a rispondere automaticamente alle chiamate (es. gestione di un BBS); infine, OH risulta acceso quando il collegamento è in atto.

A proposito, dimenticavo di dire che, a livello commerciale, si è affermato lo standard «Hayes», dal nome della casa costruttrice che per prima lo ha introdotto sul mercato statunitense. Tale standard prevede l'esistenza di un set di comandi da inviare al Modem. Tale set di istruzioni non è un vero e proprio linguaggio, bensì un insieme di macroistruzioni che semplifica, e di molto, la vita ai progettisti del software di comunicazione. In realtà, quindi, i vantaggi di questo tipo di Modem sono «trasparenti» all'utenza finale, o quasi. Infatti l'unica componente evidente è la velocità del software. Tuttavia, viste le basse velocità di trasmissione utilizzabili sulle linee telefoniche, anche questo vantaggio viene pressoché annullato!

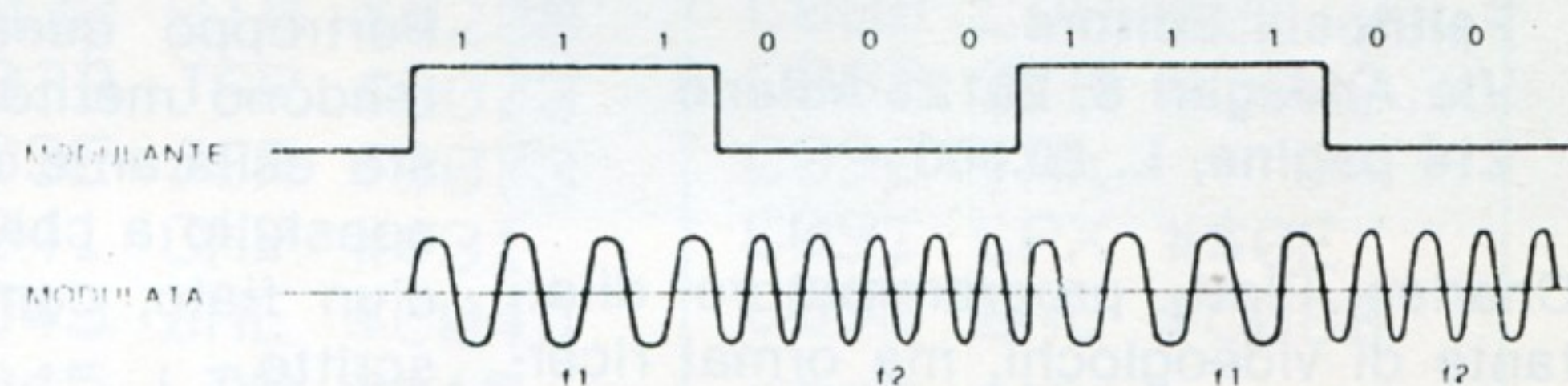


Figura 3.1 - Esempio di modulazione di frequenza.

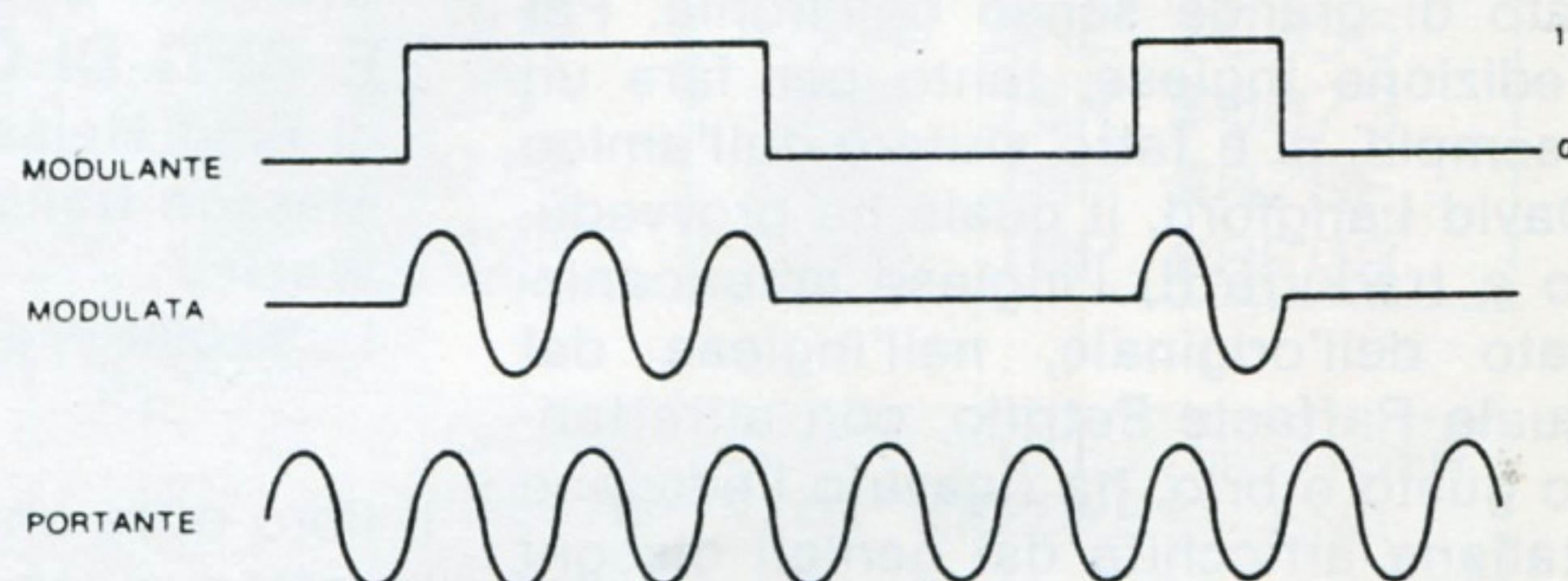


Figura 3.2 - Esempio di modulazione di ampiezza.

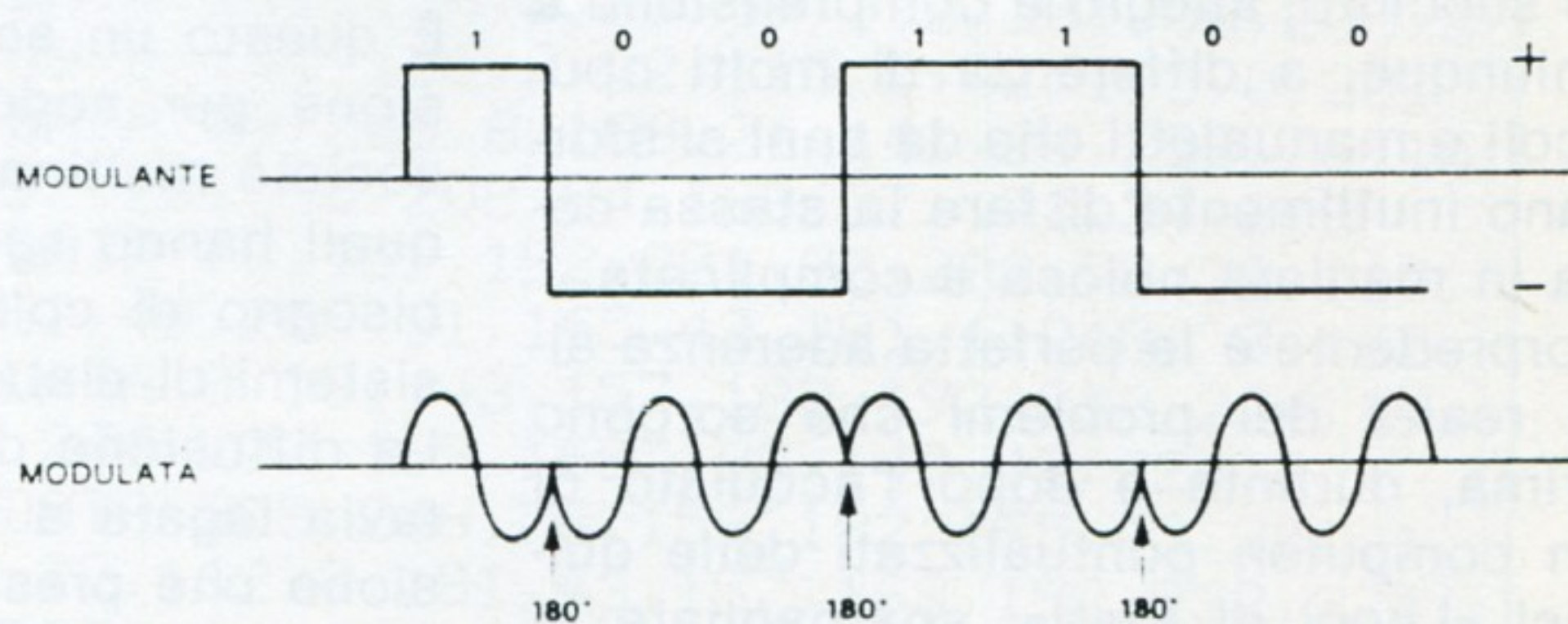
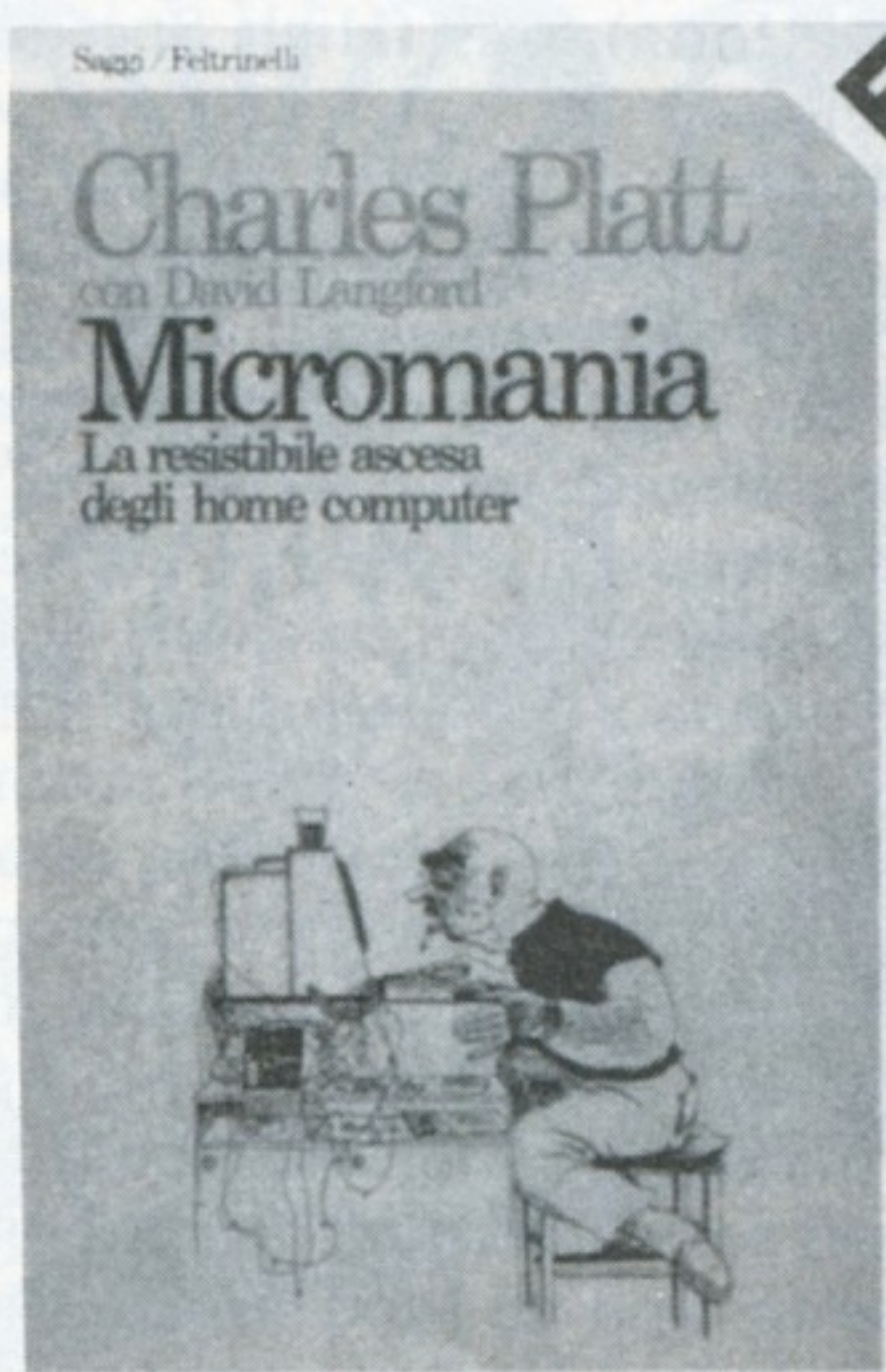


Figura 3.3 - Esempio di modulazione di fase.

LIBRI



MICROMANIA

di Charles Platt
Feltrinelli Editore
Via Andegari 6, 20121 Milano
214 pagine, L. 20.000

Charles Platt, programmatore dilettante di videogiochi, ma ormai ricercatissimo dalle case di software americane per la sua grande inventiva e quindi vicino al professionismo, è anche uno scrittore di grande vena e dotato di grande senso dell'ironia. Per l'edizione inglese, tanto per fare un esempio, si è fatto aiutare dall'amico David Langford, il quale ha provveduto a tradurre (!) l'inglese americanizzato dell'originale, nell'inglese dal quale Raffaele Petrillo, con altrettanto gusto e brio, ha ricavato l'edizione italiana arricchita dai geniali disegni di Pirro Cuniberti.

La cosa essenziale è che questo libro racconta davvero gli elementi fondamentali dell'informatica in modo molto succinto, allegro e comprensibile a chiunque, a differenza di molti opuscoli e manuali che da anni si sforzano inutilmente di fare la stessa cosa in maniera noiosa e complicata. Sorprendente è la perfetta aderenza alla realtà dei problemi che sorgono prima, durante e dopo l'acquisto di un computer, puntualizzati delle quindici «Leggi di Platt», sparpagliate in tutto il libro. Eccovi le più spassose: (a proposito delle stampanti che si inceppano proprio quando si è momentaneamente assenti)

«Quando chiedi al computer di fare

un lavoro al tuo posto, il tempo che risparmi se ne va solitamente a sorvegliare il computer per assicurarti che faccia il lavoro a dovere»; (a proposito della grande quantità di pubblicità presente nelle riviste specializzate)

«Le riviste d'informatica esistono soprattutto per dare spazio alla pubblicità. Il testo, relegato quasi sempre sulle pagine di sinistra, è presentato in modo da recare il minimo danno possibile agli inserzionisti»;

(a proposito dei word processors)

«Il problema di quasi tutti gli elaboratori di testi è che si perde metà del tempo per pensare al programma, invece di pensare a quel che si dovrebbe scrivere».

Conclusioni

Purtroppo queste poche righe non rendono merito al contenuto di questo esilarante e geniale volume che consiglio a chiunque di leggere tutto d'un fiato, come ha fatto il sottoscritto.

Il prezzo è corretto.

Fabio Scafati

SISTEMI DI COMUNICAZIONE E RETI DI COMPUTER

di Fred Halsall
Masson Italia Editore/Addison Wesley
L. 30.000

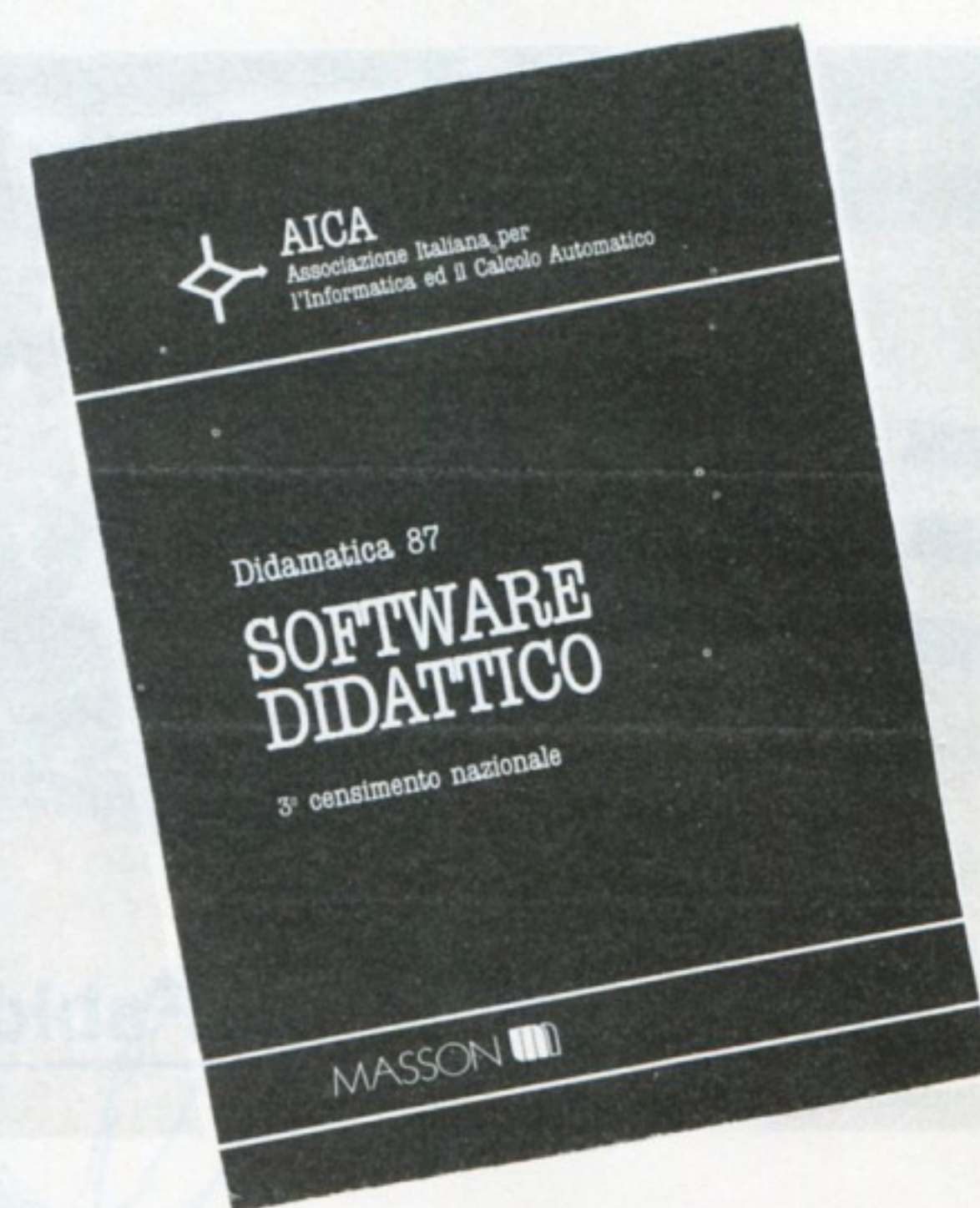
Il libro offre una moderna e autorevole introduzione ai principi e alle tecniche proprie della trasmissione dati, con particolare riguardo per reti di calcolatori e per sistemi di informatica distribuita.

È questo un settore in rapida espansione per soddisfare le richieste di società multinazionali e multisede, le quali hanno sempre più impellente il bisogno di collegare in rete i propri sistemi di elaborazione dati.

La diffusione di questi sistemi è tuttavia legata a vincoli di interconnessione che presuppongono una forma standard internazionale.

È questo uno dei temi fondamentali del volume. Inoltre il testo riporta alla fine di ogni capitolo degli esercizi adeguati.

Robert Feltoni



DIDAMATICA '87: SOFTWARE DIDATTICO

3° Censimento Nazionale
a cura dell'AICA (Ass. Ital. per l'informa.ca e il Calcolo Autom.
Masson Italia Editore
342 pagine, L. 40.000

Come molti di voi avranno già capito, «Software Didattico» dell'AICA non è un vero e proprio «libro», bensì una raccolta, sotto forma di schede informative, del software didattico circolante in Italia.

Finalmente — qualcuno penserà — abbiamo una raccolta selezionata di programmi che ci potranno essere utili nello studio o in altre attività. E invece no: la raccolta è stata realizzata attraverso la più ampia divulgazione possibile della scheda-censimento ma il carattere volontario e libero dell'adesione di chi la restituisce compilata all'AICA non consente di qualificare l'insieme dei dati raccolti. In altre parole, la descrizione di ogni programma riportata nelle schede, è quella fornita dal produttore, quindi, inevitabilmente parziale.

Perché l'AICA fa questo censimento? Le ragioni sono rimaste le due dichiarate sin dall'inizio del primo censimento:

Contribuire a far conoscere quanto viene prodotto e reso disponibile nel nostro paese, al fine di facilitare l'incontro fra gli addetti ai lavori da un lato e fra domanda e offerta dall'altro. Contribuire alla costruzione di una «conoscenza critica» di quanto viene prodotto, dato che il settore del software didattico è estremamente disomogeneo e le soluzioni proposte devono essere accuratamente studiate e valutate, prima di poter essere accettate come utilizzabili.

A chi è destinato

Agli esperti di Software Didattico per conoscere cosa si sta sviluppando in Italia da parte di altri esperti. Agli utenti docenti, studenti di scuole e università, responsabili organizzativi per scoprire che cosa è disponibile e iniziare a conoscerlo attraverso alcuni parametri significativi. Agli operatori dell'informatica che vi potranno trovare interessanti elementi relativi a un ambiente relativamente nuovo, in cui l'elaboratore si sta aprendo una strada.

Descrizione generale

Il volume contiene 292 schede compilate dagli stessi Autori che descrivono, per ogni programma, gli obiettivi didattici, gli strumenti informatici su cui funziona, la documentazione di cui è corredato, le sperimentazioni a cui è stato eventualmente sottoposto e, per molti dei programmi commercialmente disponibili, anche il prezzo.

Conclusioni

Un appunto va fatto all'indice per materie: capita infatti che manchino alcune schede, presenti invece sull'indice (quelle del CNR sugli urti); alcuni riferimenti alle pagine sono sbagliati, altri rimandano a pagine contraddistinte da un numero che è maggiore di quello dell'ultima pagina del volume.

Fabio Scafati

INFORMATICA E ALFABETIZZAZIONE

di L. Trisciuzzi

Tecnodid, Napoli, 1986

136 pagg. - L. 20.000

Anche se il sottotitolo del libro è «L'uso del computer nella scuola elementare», esso può essere usato, per una prima alfabetizzazione informatica, anche nella scuola media.

Il libro si compone di quattro parti: la prima è un'introduzione culturale al computer «Un po' di storia, L'hardware, I linguaggi, Computer e apprendimento, il programma, Avvertenze e istruzioni»: le altre tre riguardano la programmazione.

In particolare, nella seconda parte, c'è un'introduzione al BASIC dello Spectrum, mediante sette programmi didattici (Chi cerca trova. Indovina la nota musicale. Targhe d'auto. Opera-

zioni di aritmetica, Bombardiere matematico, Anagrammi per i più piccini); nella terza parte un'introduzione alla grafica ed al suono, sempre dello Spectrum, con otto programmi vari; la quarta parte è costituita da un solo lungo programma per imparare a leggere, con quattro livelli di difficoltà. I programmi possono essere adattati ad altri piccoli computer, con un minimo di conoscenza delle tecniche di programmazione e del particolare BASIC usato.

L'Autore, che è docente ordinario di Metodologia e Didattica all'Università di Firenze, ha usato un linguaggio chiaro, senza inutili tecnicismi, ed ha puntigliosamente spiegato ogni pezzo dei programmi, in modo che il Lettore non si limiti a battere passivamente i medesimi sulla tastiera, per l'effetto desiderato dall'Autore, ma impari il significato e l'uso delle varie istruzioni impiegate.

Giuseppe Festa



INTERFACCIAMENTO DEI MICROCOMPUTER

di R.C. Holland

Franco Muzzio & C. editore

240 pagine, L. 18.000

A chi è destinato?

Il testo è rivolto principalmente a coloro che desiderano conoscere le tecniche e le metodologie che permettono l'interfacciamento tra microcomputers e dispositivi esterni.

Non sono comunque, vista la semplicità con cui vengono trattati gli argomenti, richieste specifiche conoscenze di base.

Descrizione

Sono ormai disponibili molti libri che offrono un'introduzione alla tecnologia dei microprocessori.

Questi testi forniscono la descrizione dei principi fondamentali dei circuiti microelettronici e del funzionamento e della programmazione dei microprocessori.

Essi tralasciano dei modi con cui i microcomputers possono venire collegati a dispositivi esterni o a periferiche.

Questo libro esamina in dettaglio le tecnologie e i circuiti per l'interfacciamento. Tratta i principali componenti, e illustra come possono venire utilizzati sotto forma di circuito, e programmati per svolgere compiti di interfaccia.

Sono inclusi la descrizione dei metodi per il collegamento dei microcomputer a dispositivi quali stampanti, terminali video, monitor TV, display, tastiera, unità di controllo, altri computers, strumentazione di processo e viene inoltre fornita un'ampia lista di dispositivi specifici.

Vengono poi esaminati in dettaglio componenti e sistemi di interfacciamento con i microprocessori dei seguenti costruttori: Intel, Zilog, Texas Instruments, MOS Technology e con i microcomputers BBC e Apple.

È stata inclusa pure l'illustrazione dei principi di funzionamento dei microcomputers in modo che tale testo possa offrire una guida completa per apprendere le basi fondamentali della tecnologia dei microcomputers.

Nei capitoli 1, 2 e 7, ad esempio, viene descritta la tecnologia generale (circuiti logici, blocchi circuitali di base e loro programmazione), mentre il resto del libro conduce il lettore attraverso le basi e i dettagli dell'interfacciamento.

Al termine del volume si trova un Glossario che fornisce una sommaria spiegazione dei termini tecnici usati nel libro.

Giudizio

Il testo si compone di undici capitoli armoniosi e ben articolati, ognuno dei quali è corredato di numerosi schemi tecnici e disegni per rendere più facile la comprensione dell'argomento trattato.

Esso è anche dotato di una appendice contenente il famoso codice ASCII (ormai la presenza di tale codice nelle appendici di un libro sui computers è un classico) e di glossario contenente il significato dei termini tecnici usati nel testo.

Anche il costo del libro ci sembra ragionevole.

Robert Feltoni

INTRODUZIONE AL SISTEMA MS-DOS

Parte 2

di P. Ciancarini

Questa rubrica è dedicata al sistema operativo MS-DOS, quello dei personal IBM e dei cosiddetti «compatibili», che recentemente il Ministero della Pubblica Istruzione ha scelto come computer standard da usare nella Scuola Superiore.

Senza software, un computer è solamente un inutile pezzo di metallo e plastica senza vita. La sua anima sono i programmi.

Esistono due tipi di software: i *programmi di sistema*, che gestiscono le operazioni del computer stesso, ovvero sia l'interazione verso l'utente che quelle che avvengono tra i vari dispositivi, e i *programmi applicativi*, che risolvono i particolari problemi degli utenti.

Il più importante dei programmi di sistema è il **sistema operativo**, che controlla tutti dispositivi che fanno parte dell'elaboratore e fornisce la base sulla quale possono operare le applicazioni. Se chi programmi queste ultime dovesse preoccuparsi di programmare in tutti i dettagli la gestione dei dispositivi, come per esempio la lettura e la scrittura delle informazioni su un dischetto (per ciascuna operazione delle quali occorrono decine o addirittura centinaia di istruzioni di linguaggio macchina), ben pochi sarebbero i programmi a disposizione del pubblico! È il sistema operativo che si fa carico di sollevare il programmatore da tutti i compiti di routine, facilitandogli il compito di concentrarsi sul programma vero e proprio.

Il sistema operativo in effetti ha due compiti fondamentali sostanzialmente differenti. Da un punto di vista astratto, ha il compito di semplificare l'uso dell'elaboratore, dando

all'utente l'impressione di avere a disposizione una macchina molto più semplice da usare che non la macchina reale (questa macchina astratta viene spesso indicata col nome di *macchina virtuale*). Da un punto di vista più concreto, il sistema operativo deve gestire le risorse dell'intero sistema, coordinando le interazioni dei processori, delle memorie centrali o permanenti, dei dischi, delle stampanti, delle interfacce di rete, ecc...

Nello schema a fondo pagina è riassunta la gerarchia di cui si compone un sistema di calcolo. Al livello più basso vediamo l'**hardware**, che a sua volta si compone di vari livelli (non è scopo di questa serie di articoli entrare nei relativi dettagli). Come si vede, il livello base del software di sistema è costituito dal sistema operativo.

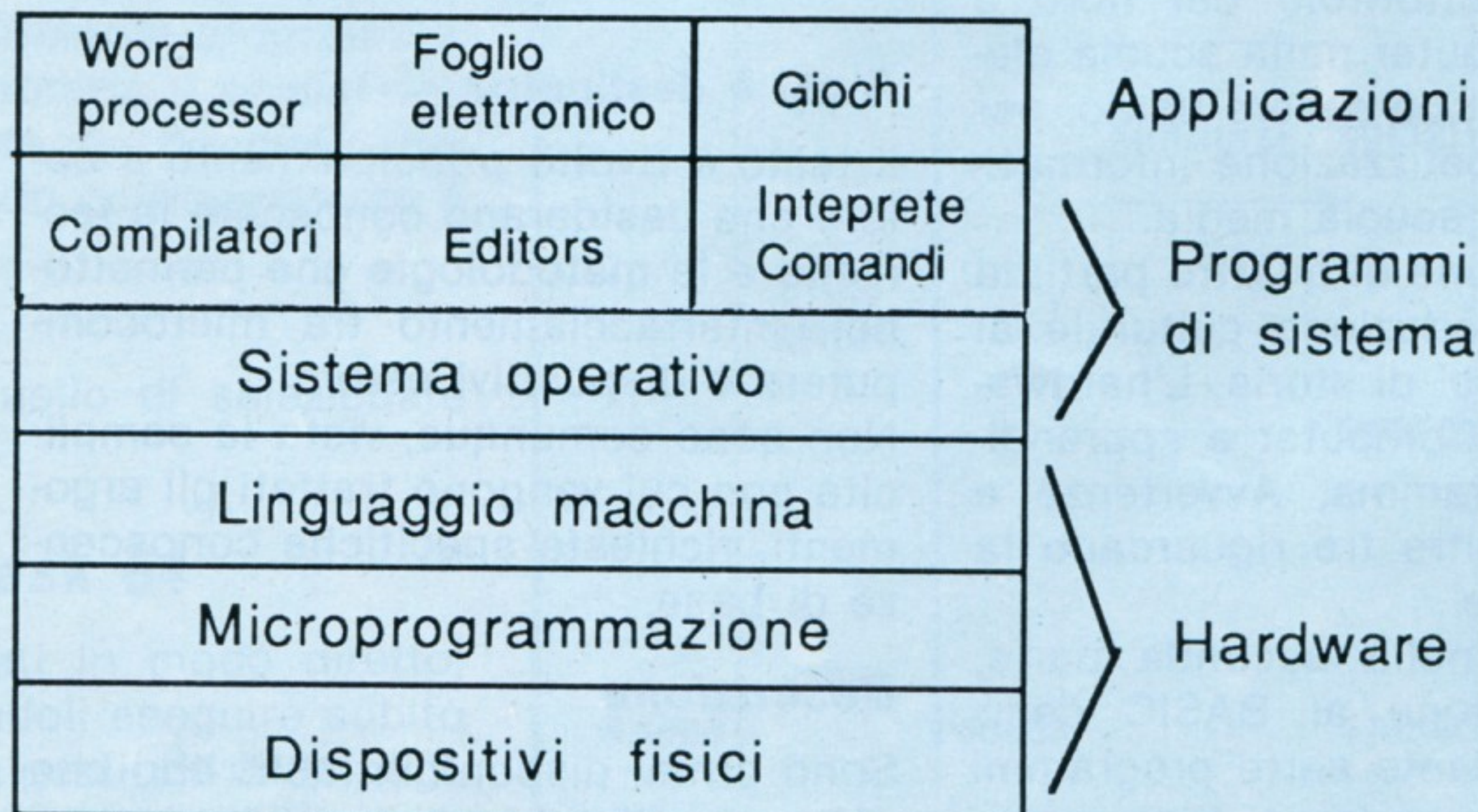
Delle due, la funzione più importante per degli utenti alle prime armi quali siamo noi è certamente la prima: occorre imparare bene quella mezza dozzina di comandi che ci permettono di sfruttare facilmente le più comuni funzionalità del nostro computer. Solo se ci addentreremo nei meandri della programmazione di sistema occorrerà diventare familiari coi dettagli della programmazione delle funzioni di gestione del sistema operativo.

I comandi principali del sistema operativo DOS sono i seguenti:

dir	elenca il contenuto del disco;
format	prepara per l'uso un dischetto vergine;
copy	copia un file da un disco all'altro;
cd	cambia directory;
del	cancella un archivio dal disco.

Prima di proseguire, va ricordato che i sistemi MS-DOS compatibili si trovano comunemente in due configurazioni: o a doppia unità per dischetti (chiamate *disk drive*), oppure a drive singolo ma accoppiato con un disco rigido (*hard-disk*). Nel primo caso le due unità si chiamano convenzionalmente **a:** e **b:**. Nel secondo caso le unità si chiamano rispettivamente : l'unità a floppy e **c:** quella a disco

Struttura gerarchica di un sistema di elaborazione



rigido. In ogni momento è attiva una sola unità; il computer segnala l'unità attiva mediante il **prompt**: inizialmente questo è

A>

il che significa che è attiva l'unità **a:**. Se il prompt fosse

B>

sarebbe attiva l'unità **b:**. Si noti che il disco col sistema va sempre messo nell'unità **a:**, a meno che non sia presente il disco rigido.

Per cambiare il disco attivo basta scriverne il nome dopo il prompt:

A> b:

cambia il disco attivo da **a:** in **b:**. Il prompt cambia di conseguenza.

Il comando dir

Come dicevamo la volta scorsa, un dischetto in formato MSDOS contiene una serie di elementi che rappresentano insiemi di informazioni. Dando il comando

A> dir

noi otteniamo l'elenco completo (detto *directory*) di questi insiemi — contenuti in questo caso nel disco **a:** —, e che gli informatici chiamano **file**.

Vediamo un piccolo directory tipico.

COMMAND	COM	15957	4-12-84	9.00a
ASSIGN	COM	772	4-12-84	9.00a
CHKDSK	COM	6468	4-12-84	9.00a

In questo elenco, la prima colonna rappresenta il *nome* dell'archivio; la seconda, chiamata **estensione**, rappresenta il *tipo*; la terza la *lunghezza in caratteri* del contenuto; la quarta e la quinta rispettivamente la *data* e l'*ora* di creazione dell'archivio (la lettera *a* finale sta per *antimeridies*, ovvero l'ora indicata appartiene alle ore del mattino; quelle del pomeriggio sono indicate col suffisso *p*, che sta per *postmeridies*). Il significato di tali informazioni dovrebbe essere ovvio anche per il lettore principiante, con l'importante eccezione di quelle contenute nella seconda colonna.

L'estensione

L'estensione è una sorta di «cognome» del file, ed ha lo scopo di stabilire il tipo delle informazioni in esso contenute. Il sistema richiede che sia assegnato obbligatoriamente a tutti i file contenenti dei programmi, cioè degli insiemi di istruzioni scritte in qualche linguaggio direttamente o indirettamente eseguibili dal calcolatore. Per esempio, un file contenente un programma in linguaggio BASIC avrà il suffisso **.BAS**; un file contenente un programma scritto in PROLOG avrà il suffisso **.PRO**, e così via (in genere il suffisso dei programmi scritti in un certo linguaggio richiama il nome del linguaggio stesso). Il suffisso non è obbligatorio se il file contiene dei dati. Spesso i programmi commerciali (es. Wordstar, DBase 3, Lotus 123, ecc.) aggiungono comunque un suffisso particolare anche a certi file di dati.

I suffissi più importanti sono quelli che caratterizzano i fi-

le di sistema, che sono quei file che contengono i programmi che costituiscono il DOS, cioè il sistema operativo stesso, e che vengono in parte caricati in memoria centrale all'accensione della macchina (si ricordi che il processore può eseguire solamente i programmi che sono residenti in memoria centrale).

Le estensioni di sistema sono quattro:

.COM sta per «*command*»: se un file ha questo suffisso vuol dire che il suo contenuto è eseguibile, è cioè un programma, che corrisponde ad un comando. Questi programmi contengono codice macchina direttamente eseguibile dal processore (si dice che sono in *memory image format*, cioè in «formato immagine di memoria»). Questo è il formato della maggior parte dei comandi esterni del sistema operativo: es.: *format*, *keybit*, *chkdsk*.

.EXE sta per «*execute*»: sono analoghi ai precedenti, ma il programma contenuto è *rilocabile*, ovvero può essere eseguito in zone differenti della memoria. In altre parole, prima di eseguirli, il sistema operativo dovrà effettuare alcune operazioni di riarrangiamento delle informazioni a sua disposizione: l'esecuzione sarà leggermente rallentata. Questo è il formato della grande maggioranza delle applicazioni commerciali.

.BAT sta per «*batch*»: contiene una sequenza di comandi di sistema. I file di questo tipo sono creati dall'utente del computer allo scopo di semplificarne la gestione. In pratica viene assegnato un nome ad una sequenza prefissata di comandi, in modo da non doverla ripetere tutte le volte.

.SYS Sta per «*sys*»: viene assegnato ai file appartenenti in esclusiva al DOS. Alcuni dei file di questo tipo sono invisibili: non vengono listati nel directory. Altre estensioni comuni sono le seguenti:

.PAS	programma sorgente BASIC
.BAS	programma sorgente PASCAL
.OBJ	codice oggetto rilocabile (non direttamente eseguibile)
.PRO	programma sorgente PROLOG
.TXT	testo
.DOC	documento
.BAK	copia di sicurezza

(Nota: un *programma sorgente* di un dato linguaggio è un testo scritto direttamente in quel linguaggio; si usa questo termine per distinguere il testo «leggibile» dal programma oggetto — cioè eseguibile —, generalmente compilato, che invece contiene codice macchina).

Possiamo a questo punto chiederci quale sia il significato dei file contenuti nel disco di sistema. I più importanti sono i seguenti:

COMMAND.COM contiene i comandi interni, cioè quelli più importanti come ad esempio *dir* e *copy*.

CHKDSK.COM consente di verificare lo stato di un disco e dei file in esso contenuti (compresi quelli invisibili).

COMP.COM
DISKCOPY.COM contiene il programma che permette di duplicare il contenuto di un disco.

EDLIN.COM contiene l'editor di sistema.

FORMAT.COM contiene il programma che permette di formattare un dischetto vergine.

Continua □

LA STRUTTURA di **AMIGA**

... ecco

AMIGA

QUESTO ARTICOLO È UNA
INTRODUZIONE ALL'USO DI
AMIGA, IL NUOVO
COMPUTER DELLA
COMMODORE, CHE SI STA
SEMPRE PIÙ AFFERMANDO
SUL MERCATO E... NEL
CUORE DEI COMMODORIANI,
COME IL DEGNO
SUCCESSORE DEL C64.

di Paolo Ciancarini

la Maserati degli Home-Computer



Possiedo ormai da qualche mese Amiga, che è stato definito, paradossalmente, sia «il più potente personal computer del mondo» che «la Maserati degli Home-computer». Vedremo che le due definizioni, apparentemente contrastanti, sono entrambe molto vere. Voglio dedicare questo articolo a quelli che, come me, hanno comprato questo computer — o hanno intenzione di farlo —, e che ne sono sempre più affascinati, ma non hanno ancora ben compreso come funziona.

I primi possessori di un Amiga sono generalmente degli ex possessori del C64 che hanno voluto fare il salto nel mondo dei personal computer. Purtroppo chi si è abituato alla spartana, ma affidabile semplicità di un home computer a 8 bit — quale il glorioso e sempre valido Commodore 64 — fa grande fatica ad entrare nel mondo estremamente complesso di un personal a 16/32 bit come l'Amiga (questa considerazione vale anche per il Macintosh e l'Atari ST, che hanno in comune con Amiga buona parte dello hardware). Per di più, come al solito, la Commodore non ha ritenuto di corredare il suo prodotto di tutti i necessari manuali di documentazione: si pensi che la collezione completa dei manuali (in edizione inglese: quella italiana non esiste) costa più di 300.000 lire.

Anche per questa ragione pensiamo di fare cosa gradita ai nostri lettori, che tutti quei soldi forse non hanno voglia di spenderli.

Per le ragioni esposte sopra, conviene cominciare questa chiacchierata paragonando l'Amiga ad una descrizione sommaria della struttura di base del C64, che molti conosceranno, anche perché si ritrova sostanzialmente immutata nelle macchine di altre marche. Il sistema operativo del Commodore 64 è di una essenzialità esemplare, ed una delle cause dirette della sua grande fortuna presso il mondo degli hobbisti. Elenchiamone le caratteristiche:

- **Sistema operativo su ROM completamente aperto, cioè noto e acces-**

sibile al programmatore anche inesperto.

- **Linguaggio BASIC residente su ROM, comprendente i principali comandi di sistema (cioè LOAD, SAVE, RUN, ECC.).**

- **Macchina monoprogrammata: in ogni momento, in memoria centrale può essere presente al più un solo programma.**

- **Grafica in alta risoluzione gestita dal processore centrale e accessibile da linguaggio macchina.**

- **Processore specializzato per il suono (non stereo), a tre voci, accessibile da linguaggio macchina.**

Il contrasto tra il C64 e AMIGA è eccezionale: si può dire che cambia tutto.

- **il Commodore 64 è una macchina unica (il 128 semplicemente «contiene» il 64, ma non lo estende). Amiga è una vera e propria linea di elaboratori in crescita costante: quest'anno sono stati annunciati due nuovi modelli.**

- **Il sistema operativo, chiamato AMIGADOS, viene interamente caricato da disco, per motivi di flessibilità e di espandibilità della linea. Occorrono, anzi, due dischi: il KICKSTART e il WORKBENCH (i nuovi modelli, AMIGA2000 e AMIGA500, hanno invece il KICKSTART residente su ROM).**

- **Apparentemente non esiste alcun linguaggio di sistema privilegiato, anche se la confezione contiene un secondo disco comprendente un interprete BASIC (della MICROSOFT[™]) abbastanza evoluto. Vedremo in realtà che il linguaggio più adatto per programmare l'Amiga (questo è vero anche per il Macintosh e l'Atari ST) è il linguaggio C, lo stesso del sistema operativo UNIX[™], cui si ispira.**

- **Amiga può eseguire più programmi contemporaneamente (questa possibilità in inglese si chiama *multitasking*).**

- **Amiga possiede vari processori speciali specializzati per la grafica e il suono stereo a 4 voci, inoltre esiste la possibilità di ottenere sintesi vocale: esiste un comando di sistema (say) che fa «parlare» il computer.**

La struttura interna del software di base di Amiga deve gestire le grandi possibilità offerte dal software ed è in effetti molto

complessa. Ci sono varie cose di cui occorre tenere conto nel tentare di spiegare il suo funzionamento. A ciascuna, come vedremo, corrisponde un modulo ben definito del sistema.

- **Interfaccia grafica amichevole guidata dal mouse. È chiamata WORKBENCH, e costituisce la grande attrattiva di Amiga. Grazie a questo programma (fornito gratuitamente all'acquisto della macchina), è possibile usare facilmente il computer, almeno per le operazioni più importanti: caricamento, esecuzione e copia dei programmi. Il principio di funzionamento è identico a quello di GEOS, il nuovo sistema operativo del C64 (residente anch'esso su disco): non occorre più ricordare i nomi dei vari comandi di sistema operativo: si fa tutto spostando il puntatore del mouse sul video, e clickando il bottone sull'icona del programma che si vuole usare.**

- **Librerie grafiche (chiamate complessivamente INTUITION). È semplice costruire programmi sofisticati dalla grafica accattivante, grazie al gran numero di primitive grafiche utilizzabili da chi sa programmare in linguaggio C.**

- **Multitasking, grazie al sistema operativo chiamato EXEC. Amiga è per ora l'unico personal computer al mondo capace di eseguire contemporaneamente più programmi.**

- **Completa apertura ed espandibilità del sistema. L'AmigaDOS, grazie alla CLI (Command Line Interface), è del tutto accessibile: non viene na-**

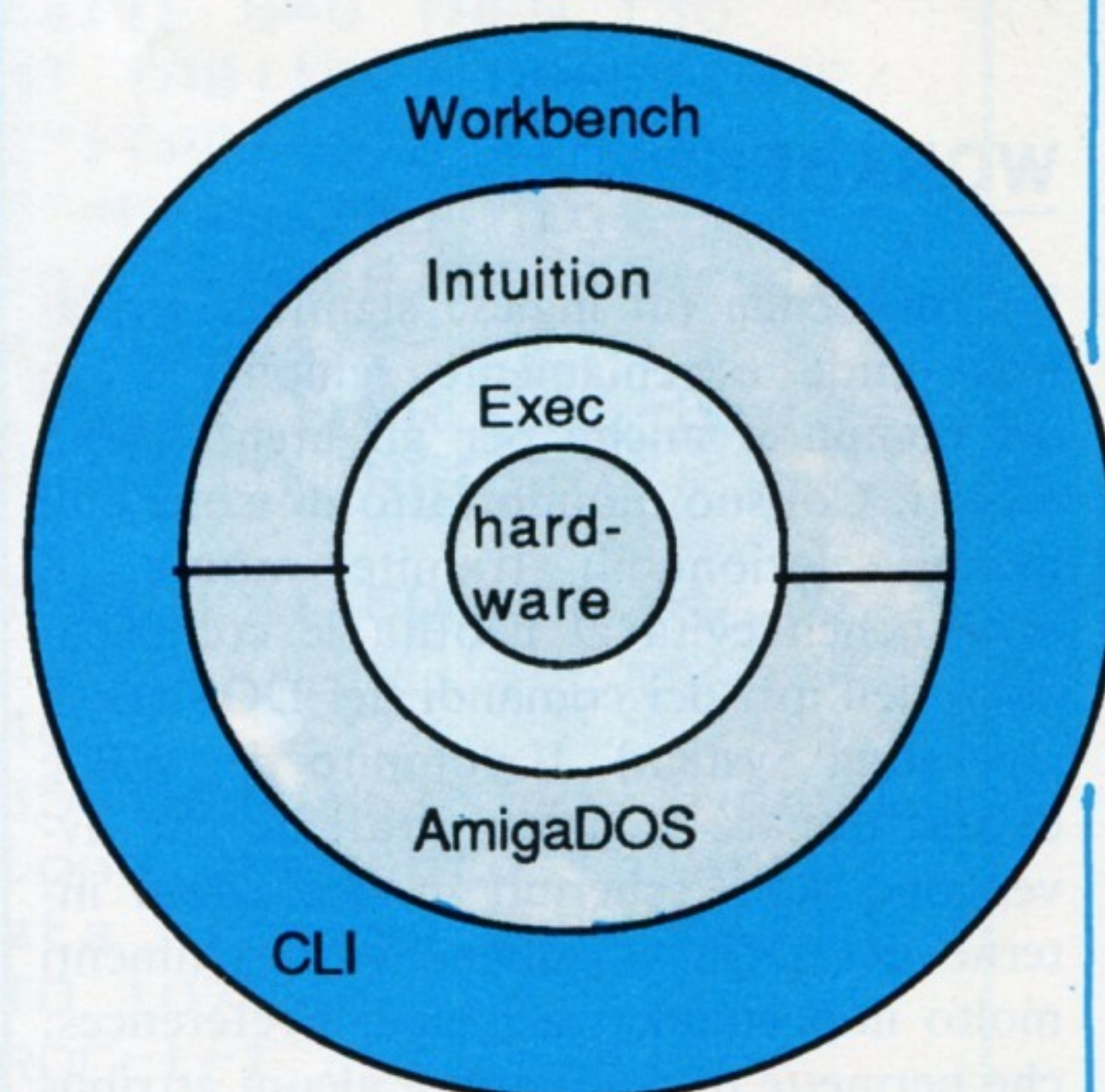


Fig. 1. Architettura di base di Amiga



AMIGA

la Maserati degli Home-Computer

scosto nulla della struttura interna del sistema operativo. Il Macintosh, per esempio, è viceversa completamente chiuso. Questo rende l'Amiga una macchina molto più adatta per l'utente inesperto per imparare come è fatto dentro un sistema operativo complesso.

- **Autoconfigurazione:** il sistema operativo riconosce automaticamente la presenza di uno o più disk drive, delle eventuali espansioni di memoria, di eventuali coprocessori anche con differenti sistemi operativi (per esempio il SIDECAR, l'emulatore PC IBM).

Per chiarire meglio la struttura del sistema operativo, si osservi la figura seguente, che viene chiamata «diagramma a cipolla», e che serve per rappresentare la gerarchia di relazioni intercorrente tra i vari componenti del software di base.

All'acquisto, il computer viene fornito con due dischi di sistema (più altri due, contenenti rispettivamente l'interprete BASIC e alcuni programmi dimostrativi. Il disco *Kickstar* contiene gli «strati» Exec a AmigaDOS della cipolla. Il disco *Workbench* contiene lo strato più esterno. Esaminiamo adesso più da vicino le funzioni dei vari moduli del sistema Amiga, a cominciare da quelli immediatamente visibili all'utente.

WORKBENCH

Il Workbench (in inglese significa scrivania) rende estremamente semplice l'uso del computer anche per gli utenti meno esperti. Col suo mondo fatto di icone e di menu selezionabili tramite mouse, il Workbench evita al neofita le complicazioni dell'uso dei comandi del DOS (Disk Operating System). Il compito del WB è di trasformare le azioni grafiche che avvengono sullo schermo in operazioni interne del DOS. Contiene vari strumenti molto importanti, tra i quali Preferences, che permette di configurare alcuni attributi del sistema (colore, stampante, orologio interno, ecc.), Icon Editor, che permette

di personalizzare le icone, DiskFormat e DiskCopy, per gestire i dischetti. Raccomandiamo ai lettori di studiare approfonditamente nel manuale contenuto nella confezione la descrizione di tutte le funzionalità del Workbench. Ci si familiarizzi soprattutto con le nozioni di sfondo, di finestra e di gadget.

CLI (Command Line Interface)

Chi usa solamente il Workbench perde una buona parte della potenza che Amiga è capace di sviluppare. Ci sono cose che solo direttamente i comandi DOS possono fare. A questo scopo il WB contiene una icona che rappresenta un sottoambiente capace di trasformare Amiga in una macchina convenzionale, in cui i comandi che si possono dare sono quelli dell'AmigaDOS, e sono circa mezzo centinaio (NOTA: la versione 1.2 contiene 5 comandi in più). I più importanti sono:

- dir** per listare il directory corrente
- cd** per cambiare directory
- endcli** per uscire dalla CLI

AMIGADOS

È il sistema operativo vero e proprio, cioè il programma che si cura di gestire tutte le funzioni del computer: elaborazione centrale, memoria interna, memoria esterna, video, audio, dispositivi di stampa, ecc. È di concezione moderna, e somiglia molto ad un sistema operativo «grande» e famoso: UNIX. La caratteristica più interessante è il sistema di archiviazione dei file (file system), organizzato in modo gerarchico.

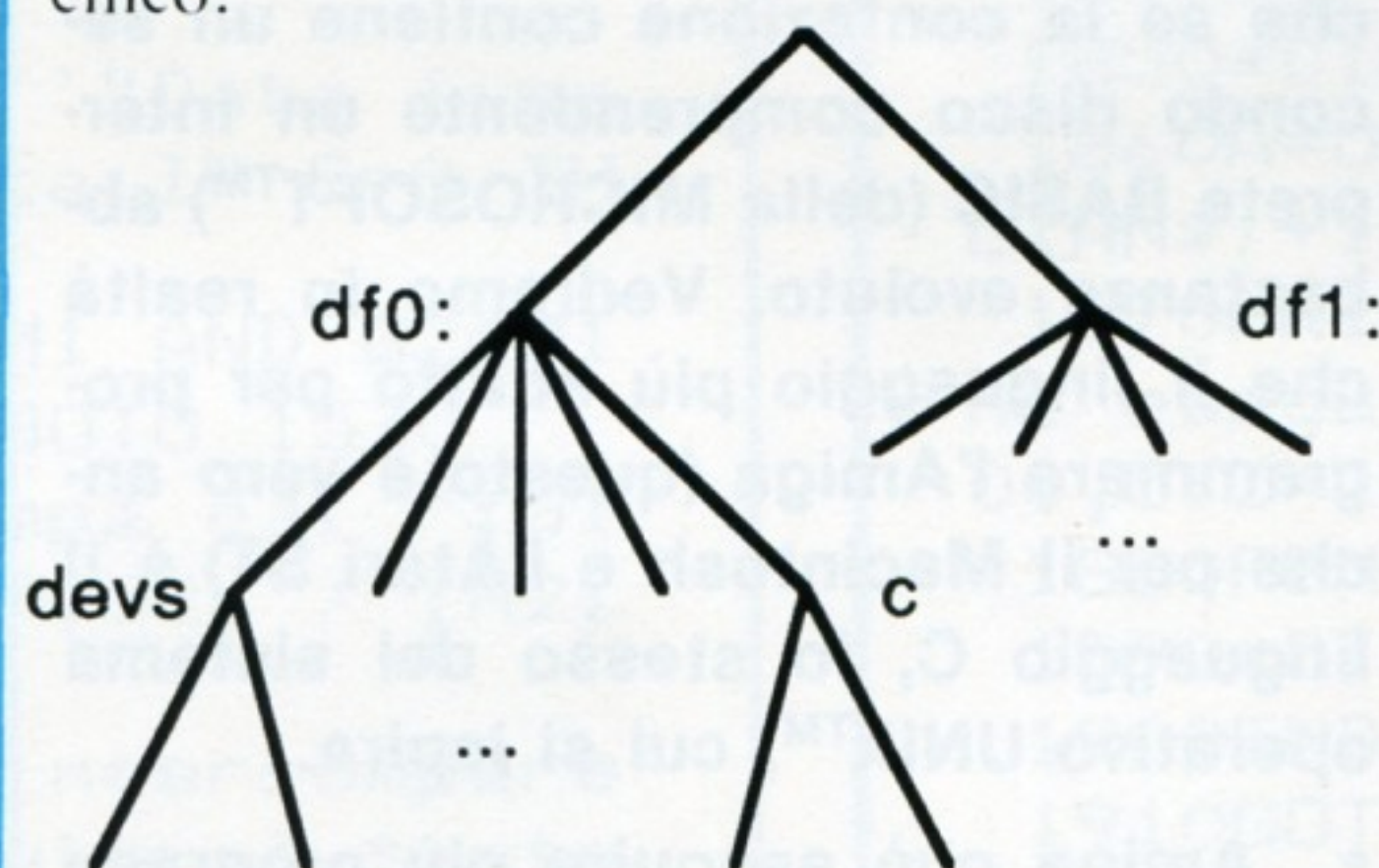
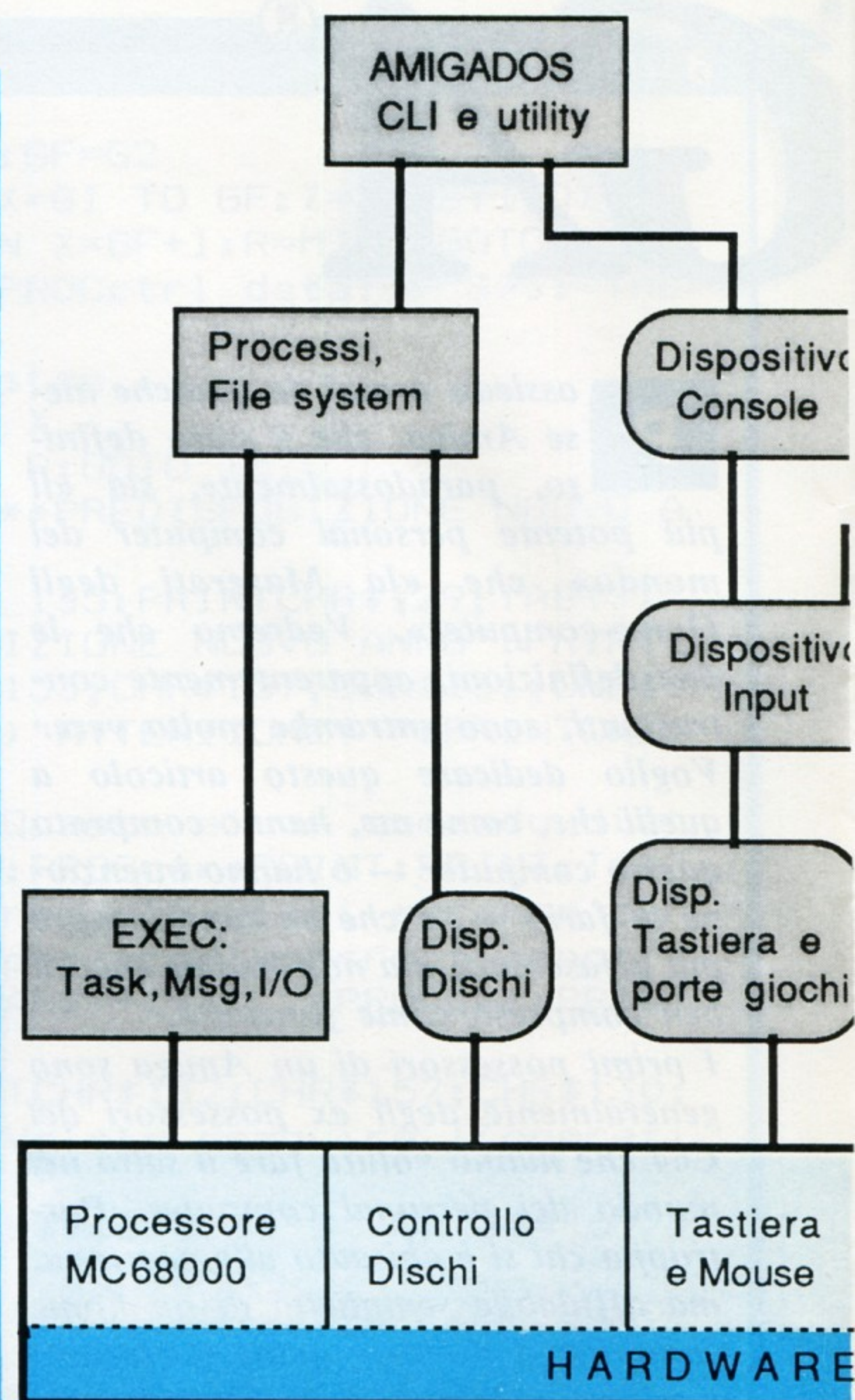


Fig. 2. Il File System di Amiga

INTUITION

A questo modulo è affidata l'interpretazione delle scelte che facciamo quando muoviamo il puntatore del mouse sulle varie icone e opzioni dei menu, e premiamo il bottone. È formato da un insieme di programmi in linguaggio macchina che permettono al programmatore di sfruttare



nei suoi programmi tutte le capacità del sistema. Per usare **Intuition**, occorre saper programmare in C e conoscere il dizionario di sistema, ovvero l'elenco dei nomi di tutte le routine di Intuition.

EXEC

AmigaDOS è un sistema operativo con multitasking. Il compito del modulo EXEC (che costituisce il contenuto nel misterioso disco di Kickstart) è proprio quello per gestire tutte le complicazioni che derivano dal fatto che il sistema deve occuparsi dell'esecuzione concorrente di più processi (un processo non è altro che un programma in esecuzione). Il problema principale deriva dal fatto che processi differenti possono competere per l'uso degli stessi dispositivi (disk drive, stampante) che però possono servirne uno alla volta. Occorre quindi che il sistema operativo si faccia carico di redimere tutti i conflitti che sorgono durante la vita dei processi.

EXEC rappresenta il cuore del sistema operativo, ciò che gli informatici chiamano il *nucleo* (in inglese, il *kernel*): le sue routine vengono usate sia dalla CLI che dai programmi di Intuition. Chiaramente è uno dei moduli più interessanti ma anche più complessi da studiare.

Lo schema seguente precisa e riassume le

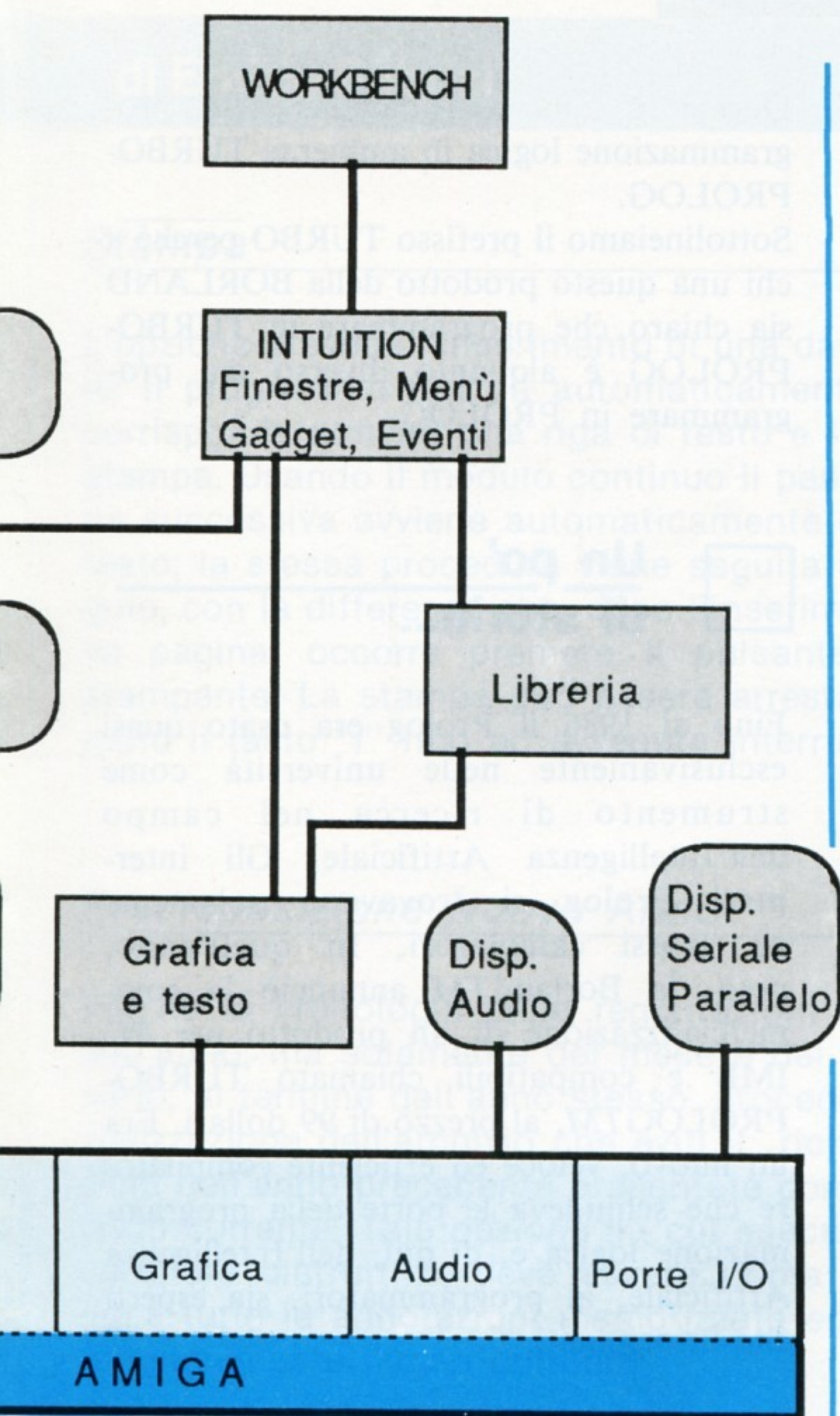


Fig. 3. L'architettura del software di base di Amiga

relazioni intercorrenti tra i vari componenti del sistema. Si noti il gran numero di *device* (dispositivi): sono programmi che gestiscono ciascuno una risorsa privilegiata del sistema, cioè per esempio un disk drive, la tastiera, la porta stampante, ecc. Queste risorse debbono essere attentamente sorvegliate perché se due processi se ne impossessassero contemporaneamente potrebbero capitare dei malfunzionamenti.

Le librerie

Il disco Workbench che ci viene consegnato all'acquisto di Amiga contiene più di cento file. Eppure le icone contenute nella schermata di base (comprese quelle nascoste nei cassette) non superano la ventina. Per vedere tutti i file «nascosti», procedere così: dal Workbench invocare l'icona della CLI e scrivere il comando

dir

Viene mostrato l'elenco completo dei directory e dei file contenuti nel disco. I nomi di molti di questi ultimi finiscono con le lettere *.info*. Spiegheremo più avanti il loro significato.

c

Contiene tutti i file corrispondenti ai comandi attivabili da CLI. Quando scriviamo un comando, l'interprete DOS cerca prima nel directory corrente: se non trova un file eseguibile corrispondente al nome del comando, cerca nel directory *c*. Si noti che la versione 1.2. del WB contiene alcuni comandi nuovi, tra i quali citiamo *addbuffers* e *mount*.

l

Contiene una serie di funzioni di libreria che estendono il DOS. Per esempio, il file *RAM-Handler* controlla il disco RAM (dispositivo disco virtuale che utilizza la memoria centrale allo scopo di velocizzare le operazioni di accesso ai file).

devs

Contiene il software di gestione dei vari dispositivi che sono usati da Amiga. Quando un programma deve usare un dispositivo, ad esempio il drive oppure la stampante, invoca la routine di sistema *OpenDevice*, che cerca il programma corrispondente in questo directory (se già non è stato usato). Si noti che contiene il directory *printers*, con i file di tutte le stampanti usabili con Amiga (selezionabili mediante lo strumento Preferences del WB). La versione 1.2. contiene anche il directory *keymap*, che permette di configurare la tastiera in formato differente da quello standard (a proposito, evitare di acquistare un Amiga avente una tastiera italiana: richiedete SEMPRE la tastiera originale americana, ed eviterete un sacco di guai!).

s

Viene usato per memorizzare i file-batch, che sono sequenze programmate e prefissate di comandi DOS. Quando EXECUTE viene invocato avendo come argomento un certo file, quest'ultimo viene cercato nel directory corrente; se non c'è si cerca in *s*. Si noti che inizialmente in questo directory si trova un solo file chiamato *startup-sequence*: contiene la sequenza di comandi che viene eseguita all'accensione del calcolatore, quando viene inserito il disco WB.

fonts

contiene le definizioni dei vari insiemi di caratteri tipografici (chiamati *fonti tipografiche*) che Amiga può usare per scrive-

re del testo. Quando un programma deve usare una nuova fonte tipografica viene invocata la routine di sistema *OpenFonts*, che controlla se la fonte è già presente in memoria. Se non c'è, viene scandito questo directory alla ricerca della fonte. Le fonti di base sono 6, ma recentemente molte ditte indipendenti hanno prodotto varie decine di fonti originali.

libs

Contiene le librerie di estensione del sistema. Funzioni tipiche memorizzate in questo directory sono *translator.library*, che serve per la sintesi vocale di testi, *diskfont-library*, per caricare e scaricare le fonti residenti su disco, *icon.library* e *info.library*, per matematica in precisione doppia, *mathieeesingbas.library*, per la matematica in precisione semplice, *math-trans.library*, per le funzioni matematiche trascendenti.

t

Inizialmente è vuoto. Viene usato da certi programmi (come per esempio gli editor di sistema) come contenitore di file temporanei.

Conclusioni

Amiga è un computer giovane, che senza dubbio crescerà nel tempo: vengono annunciati ogni mese nuovi dispositivi hardware (soprattutto estensioni di memoria e dischi rigidi) e decine di nuovi programmi. A chi ne consigliamo l'acquisto? Per ora, finché non verrà completato il parco dei programmi commerciali disponibili, lo vediamo adatto soprattutto a tre categorie di utenti, che lo useranno come:

□ **Elaboratore grafico sofisticato.** La RAI già usa questo computer come supporto di molte trasmissioni basate su grafica digitale. Le possibilità nel campo (chiamato *desktop video*) sono enormi.

□ **Macchina per giocare;** la perfezione dei videogiochi di Amiga non ha sin da adesso eguali sul mercato. Da notare soprattutto la serie Cinemaware della Mindscape (primi titoli: Defender of the Crown, Sinbad, SDI, Dejavu, Uninvited), che offre un'anteprima di ciò che sarà il futuro del gioco computerizzato.

□ **Macchina per imparare l'informatica.** La struttura del sistema operativo è quella che si imporrà sempre più nel futuro come quella del nucleo di qualsiasi computer: ricordiamo lo stretto legame di parentela che esiste tra AmigaDOS e UNIX. La valenza didattica di questo elaboratore è quindi grossissima.

Prolog e Turboprolog

Come usare il TURBOPROLOG

di Paolo Ciancarini



Il Prolog può essere appreso facilmente come primo linguaggio di programmazione. Lo presentiamo sia ai dilettanti d'informatica, sia agli esperti di qualche altro linguaggio che vogliono padroneggiare rapidamente questo nuovo strumento di programmazione.

Questa è una introduzione alla programmazione logica in ambiente TURBOPROLOG.

Sottolineiamo il prefisso TURBO perché a chi una questo prodotto della BORLAND sia chiaro che programmare in TURBOPROLOG è alquanto diverso dal programmare in PROLOG.



Un po'
di storia...

Fino al 1986 il Prolog era usato quasi esclusivamente nelle università come strumento di ricerca nel campo dell'Intelligenza Artificiale. Gli interpreti Prolog si trovavano solamente su grossi calcolatori. In quell'anno, però, la BorlandTM annunciò la commercializzazione di un prodotto per PC IMB e compatibili chiamato TURBOPROLOGTM, al prezzo di 99 dollari. Era un nuovo, veloce ed efficiente compilatore che schiudeva le porte della programmazione logica e, di qui, dell'Intelligenza Artificiale, ai programmatori, sia esperti che principianti.





La differenza si riassume in due considerazioni fondamentali:

1) Il **TURBOPROLOG** è molto di più che un linguaggio: è un ambiente integrato di programmazione estremamente ricco che permette di creare con facilità programmi efficienti anche molto complessi.

2) Il **TURBOPROLOG** non è un vero Prolog. Anche se ne condivide la filosofia generale, la sua sintassi e la sua semantica si discostano di molto dallo standard di fatto, che è quello stabilito dall'Università di Edimburgo.

Il **TURBOPROLOG** costituisce forse il primo esempio economico di ambiente integrato e flessibile di programmazione, che sia implementato su personal computer IBM (o compatibile) e che sia alla portata dell'utente non esperto. Oggi esiste un largo consenso sul fatto che l'ambiente di programmazione influenza significativamente la produttività del programmatore, e non solamente sotto l'aspetto quantitativo. Per questo motivo descriveremo innanzitutto le potenti funzionalità del sistema, invece di addentrarci immediatamente nei meandri della programmazione logica.

La seconda considerazione potrà forse stupire, ma è profondamente vera. Il **TURBOPROLOG** costituisce un riuscito esempio di compromesso tra il paradigma tradizionale della programmazione algoritmica o procedurale, e quello innovativo e affascinante della programmazione logica o dichiarativa.

Come in tutti i compromessi, si rinuncia a qualcosa per ottenere qualche altra cosa. Nel caso del **TURBOPROLOG**, la BORLAND ha rinunciato ad implementare la piena potenza espressiva del Prolog per creare un linguaggio ibrido sostanzialmente simile al Pascal. Questo è certamente un fatto positivo per quei lettori che, provenendo da esperienze di programmazione procedurale in qualche linguaggio a blocchi, vogliono avvicinarsi al mondo della programmazione logica. È d'altro canto certamente negativo per chi cercasse di trovare nel **TURBOPROLOG** una implementazione del Prolog standard. Come è noto, uno standard ufficiale del linguaggio Prolog non esiste, ma è sufficientemente diffuso uno standard *de facto*, quello descritto nel libro di Clocksin e Mellish «*Programming in Prolog*», noto come C&M Prolog oppure Prolog di Edimburgo.

Come dicevamo, il **TURBOPROLOG** è sostanzialmente un linguaggio nuovo con caratteristiche comuni, da una parte, al Prolog e, dall'altra, al PASCAL oppure al C.

La caratteristica che più lo differenzia dal

Prolog puro è il sistema di tipi. Inoltre mancano del tutto i predicati di metalivello: l'unico presente è il costruttore di insieme *findall*.

Anche l'unificazione è parzialmente limitata. Per esempio, in C&M Prolog il predicato

$a(X + Y, Z).$

unifica col predicato

$a(7 + 5, K).$

legando X a Y a 5. Come vedremo, in **TURBOPROLOG** non è invece permesso usare espressioni di questo tipo, né si possono dichiarare nuovi operatori. È vietato persino innestare nel corpo delle clausole arbitrariamente più livelli di congiunzioni *and* e *or*.

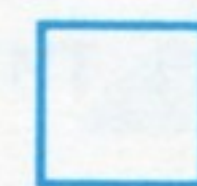
Il **TURBOPROLOG** segue rigidamente la sintassi delle clausole di Horn. La mancanza più grave è quella dei principali predicati di metalivello, che combinata con l'introduzione di vincoli sull'unificazione limita alquanto la potenza espressiva del linguaggio per ciò che riguardale manipolazioni simboliche.

Va notato che queste caratteristiche negative sono frutto di una scelta ben precisa da parte dei progettisti del **TURBOPROLOG**. Le note positive sono che tali vincoli fanno sì che il **TURBOPROLOG** sia un linguaggio il cui codice compilato è estremamente efficiente. Per di più, non si è ancora raggiunta la massima ottimizzazione: secondo la Borland la differenza di efficienza tra la versione 1.0 e la attuale 1.1. raggiunge il 400%: ciò fa sospettare che le prossime versioni saranno ancora più veloci. Inoltre, mentre in Prolog normale scrivere programmi più lunghi di 100 righe diventa un'attività estremamente soggetta ad errori, a causa della mancanza di struttura intrinseca dei programmi, il **TURBOPROLOG** permette una forma di programmazione modulare piuttosto utile ed efficace. Esistono infine alcune primitive grafiche molto potenti che permettono di sfruttare le possibilità offerte dallo hardware nella costruzione di sofisticati sistemi a finestre. Alcuni predicati di interfaccia col DOS e col sistema dei file danno al sistema una grande flessibilità d'uso persino in applicazioni di sistema operativo. È inoltre possibile costruire programmi complessi componendo insieme parecchi moduli creati separatamente. Per di più, i programmi possono essere compilati e riversati direttamente su disco in modalità .COM oppure .EXE, il che permette sia di eseguirli da DOS senza invocare il sistema **TURBOPROLOG**, sia di richiamarli tramite programmi scritti in altri linguaggi.

Una delle caratteristiche più importanti che distingue il **TURBOPROLOG** dalle

altre implementazioni del Prolog è l'ambiente di programmazione, e soprattutto la sua componente chiamata *interfaccia utente*. Con questa espressione si indica quella parte di un sistema software che si occupa di gestire l'interazione con il programmatore e/o l'utente. L'evoluzione delle tecniche di ingegneria del software sta portando alla progettazione di interfacce sempre più evolute, capaci di alleggerire notevolmente le fatiche della programmazione. Il **TURBOPROLOG** compie un passo piuttosto deciso in questa direzione, perché mette a disposizione un insieme integrato di strumenti allo stesso tempo potente e facile da usare.

Il ciclo di creazione dei programmi, che prevede le tre fasi di *editing* — *compilazione* — *esecuzione*, è grandemente velocizzato e semplificato grazie alla struttura del sistema. In questo capitolo descriviamo le funzionalità più importanti dell'ambiente **TURBOPROLOG**.

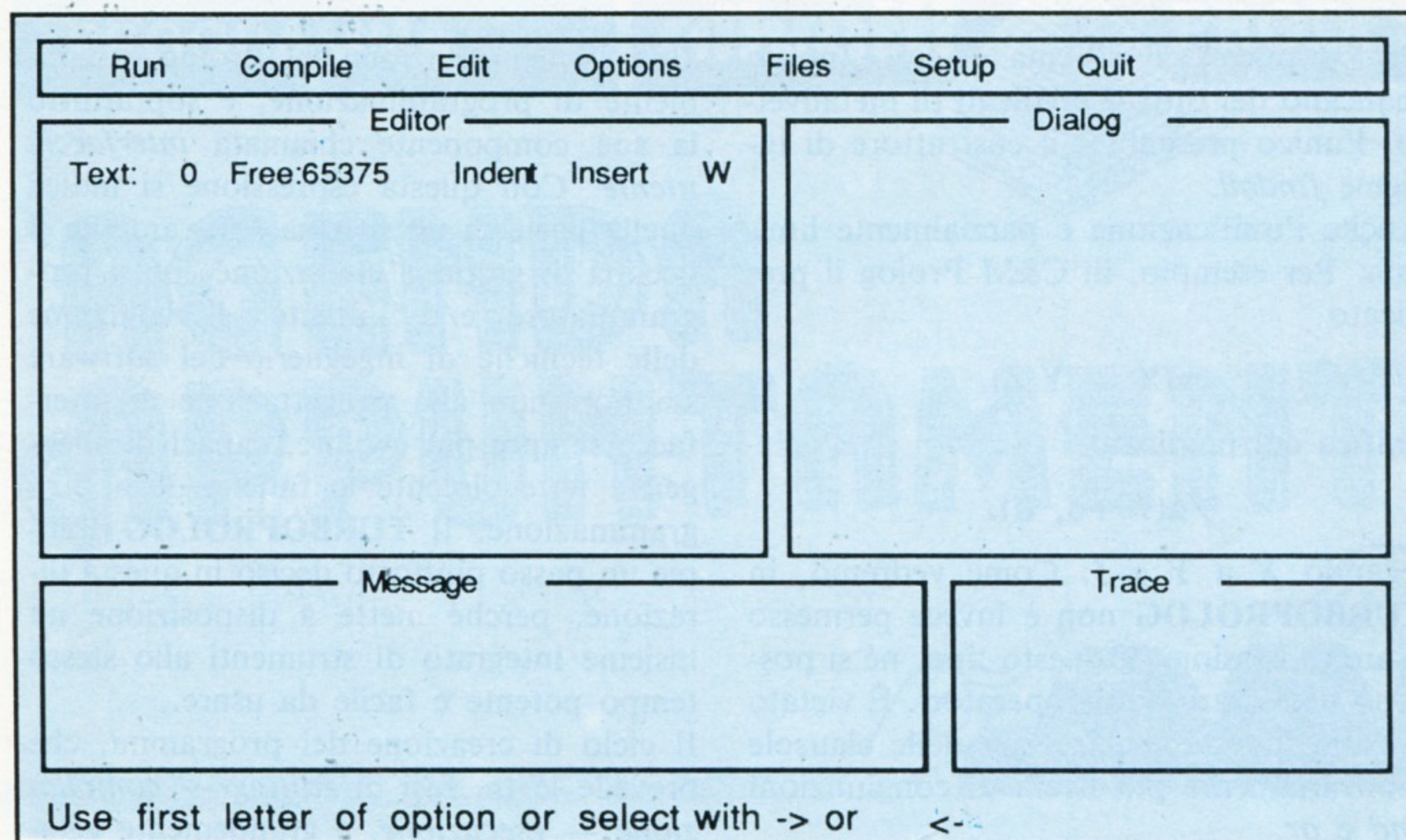


A cosa serve l'ambiente

Un ambiente di programmazione è uno strumento che consente al programmatore di lavorare comodamente e confortevolmente alla costruzione dei suoi programmi. Ciò implica che parte dei compiti di routine che di solito sono a cura del programmatore, vengono svolti dal sistema più o meno automaticamente. Chi ha programmato in codice assembler sa quanto sia importante poter usare una catena di strumenti chiamati *assemblatore*, *linker*, *loader*. In questo caso potremmo parlare di embrione di ambiente, in quanto tali strumenti costituiscono un ausilio alquanto rozzo alla programmazione di sistemi software complessi. Da questo punto di vista, il principale vantaggio di un linguaggio ad alto livello, rispetto alla programmazione in assembler, è dato dalla possibilità di controllare automaticamente la forma dei programmi prima di eseguirli, grazie ad uno strumento chiamato *analizzatore sintattico*. Sono possibili anche altri controlli, se l'ambiente fornisce gli strumenti adeguati. Praticamente ogni linguaggio moderno offre un ambiente più o meno sofisticato. Ad esempio, un interprete BASIC contiene normalmente un piccolo editor di schermo, non molto sofisticato ma adatto alla stesura dei programmi.

Un ambiente evoluto renderà disponibile un numero di strumenti ben maggiore. Gli strumenti che compongono un ambiente





L'interfaccia TURBOPROLOG

tipico sono i seguenti (ma ne possono esistere molti altri):

- l'*editor*, che serve per scrivere e visualizzare il programma;
- l'*interprete*, che serve per eseguire direttamente il testo del programma;
- l'*analizzatore sintattico*, che serve per controllare la forma del programma;
- l'*analizzatore dei tipi*, che effettua alcuni controlli semantici sul programma;
- il *debugger*, che serve per analizzare il significato del programma;
- il *generatore del codice oggetto*, che traduce il programma in linguaggio macchina;
- il *linker*, che permette di comporre programmi differenti creando un unico programma;
- il *sistema di archiviazione*, che serve per memorizzare sia i programmi sia i documenti che li riguardano.

Di solito un ambiente Prolog minimale per il sistema MS-DOS offre solamente l'interprete, il debugger e qualche funzione di archiviazione su memoria permanente; l'interfaccia col sistema operativo generalmente è del tutto assente. Il **TURBOPROLOG** fornisce invece tutto questo insieme di strumenti, integrandoli all'interno di una interfaccia utente estremamente amichevole e facile da usare, e per di più permette di usare i programmi Prolog in combinazione con quelli di altri linguaggi.

Usare il TurboProlog

A meno di non voler usare l'unità di memoria permanente a disco rigido, non è necessaria alcuna operazione preventiva di installazione per usare il **TURBOPROLOG**.

Inseriti i due dischetti trovati nella confezione nelle relative unità del calcolatore, digitare

A > prolog

Al termine del caricamento viene mostrata una schermata d'apertura contenente il messaggio di copyright della Borland, insieme agli assegnamenti iniziali di vari file che verranno definiti ed usati dal sistema durante la creazione e la eventuale successiva compilazione del programma. Premendo il tasto «SPAZIO» si entra in ambiente TURBOPROLOG. Lo schermo è a questo punto suddiviso graficamente in 6 parti.

Menu:

È la riga in alto: permette di selezionare i comandi principali dell'ambiente. Le opzioni sono sette; da sinistra a destra: Run, Compile, Edit, Options, File, Setup, Quit. Le varie opzioni vengono selezionate o battendo l'iniziale corrispondente, oppure mediante i tasti cursore in combinazione col tasto «RETURN».

Editor:

La finestra grande a sinistra: è il sottoambiente che permette di creare e modificare il testo dei programmi. Viene attivata dal Menu mediante l'opzione Edit.

Dialog:

La finestra grande a destra gestisce l'interazione Prolog. È il sottoambiente mediante il quale si fanno le interrogazioni e nel quale compaiono i risultati dei programmi. Viene attivata dal Menu mediante l'opzione Run.

Message:

La finestra in basso a sinistra. Qui compaiono i messaggi del sistema in risposta alle nostre operazioni (ed eventualmente i messaggi di errore, se sbagliamo qualcosa).

Trace:

La finestra piccola in basso a destra permette di osservare lo svolgersi dell'esecuzione di un programma.

Riga di Help:

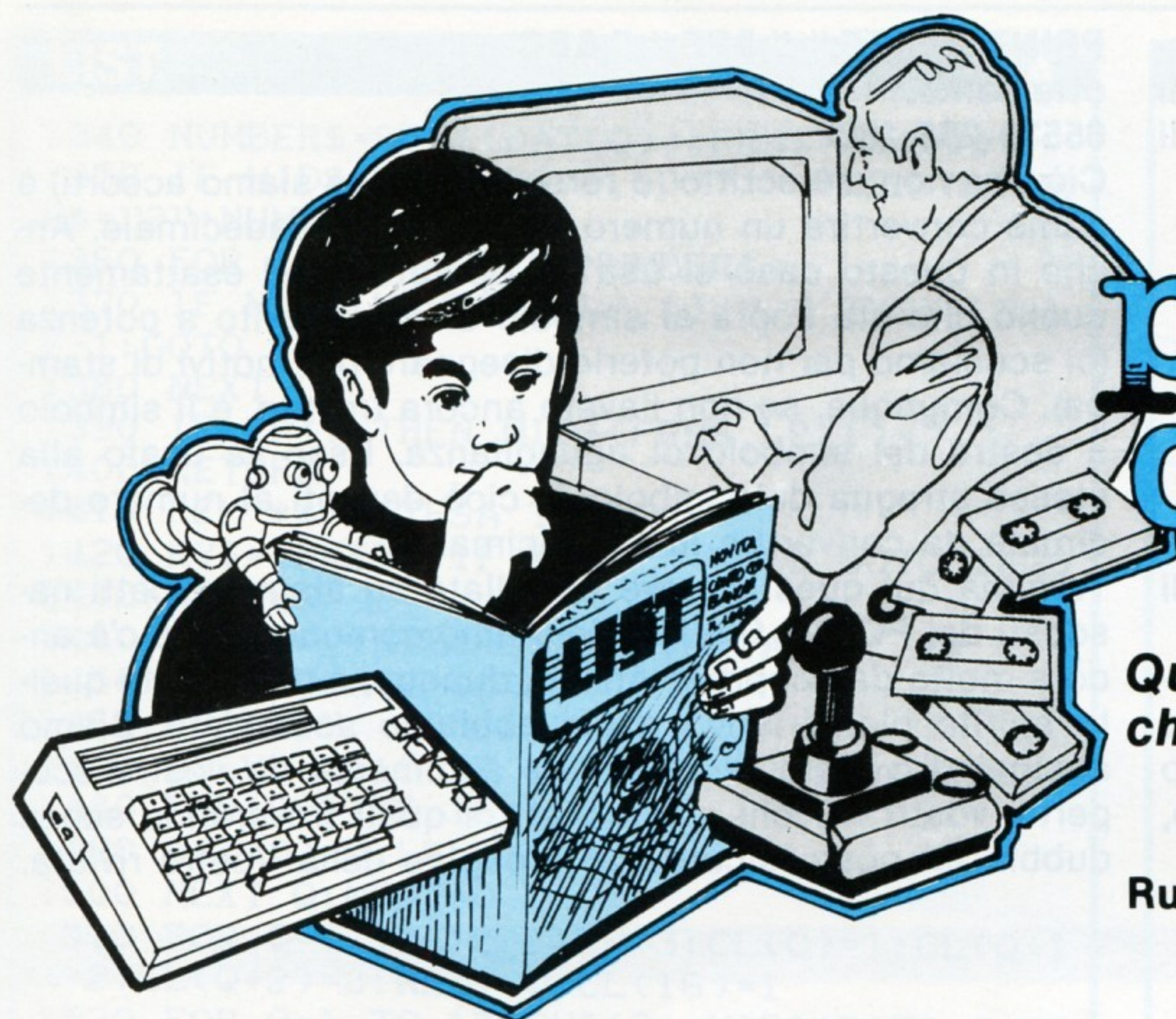
È una riga che contiene un riassunto dei comandi principali usabili nel sottoambiente attivo.

La funzione delle opzioni e dei sottoambienti va compresa tenendo presente quali sono le principali operazioni che compie chi crea un programma. Il ciclo base di creazione, esecuzione e correzione di un programma **TURBOPROLOG** è il seguente:

- 1) **Entrare nell'Editor e scrivere il testo del programma.**
- 2) **Uscire dall'Editor e salvare il programma mediante l'opzione Files del Menu.**
- 3) **Compilarlo mediante Compile (seconda opzione del Menu).**
- 4) **Eseguirlo, attivando la finestra di Dialog tramite l'opzione Run del Menu.**
- 5) **Se necessario, correggerlo usando l'Editor e ricominciando daccapo il ciclo.**

Inizialmente è possibile solamente selezionare una qualsiasi opzione del Menu. Se viene attivata una delle quattro finestre centrali, il corrispondente sottoambiente attivo viene segnalato dal cambiamento della luminosità della sua cornice. Va subito ricordato un tasto molto utile per uscire da qualsiasi sottoambiente e tornare al Menu: è quello di «ESCAPE».

Nel prossimo numero descriveremo dettagliatamente i vari sottoambienti con i comandi relativi.



per saperne di più...

Quello che non dicono i manuali e quello che nei manuali è sbagliato

Rubrica aperta anche alla collaborazione dei lettori

OLIVETTI PRODEST PC 128 S ...dentro la memoria

Una delle operazioni senz'altro più divertenti, quando si ha tra le mani un nuovo computer, è quella di andare a ficcanasare direttamente nella memoria della nostra macchina, magari per vedere come vengono memorizzati i programmi o le variabili, come viene gestito il display file, ecc. Per far ciò, però, bisogna conoscere due importanti parole del BASIC: una è il comando POKE, con il quale si scrive nella memoria, e l'altra è una funzione, PEEK, con la quale si legge una locazione. Molti dialetti BASIC usano una sintassi del tipo:

POKE Locazione, Valore

per alterare il valore della Locazione memorizzandoci il Valore; per la funzione invece:

PRINT PEEK (Locazione) oppure **PRINT PEEK** Locazione
per leggere il contenuto della Locazione.

Chi era già a conoscenza di ciò sarà rimasto deluso quando, sedutosi davanti al proprio PC 128 S con spirito più da pioniere che da programmatore, si è accorto che il manuale in dotazione non riporta comandi né sintassi delle citate «parole». In realtà il BASIC BBC usa un modo un po' particolare per gestire le operazioni sulla memoria. La prima operazione che prenderemo in considerazione è la **modifica del valore di una cella di memoria**. Ciò viene fatto con i seguenti operatori:

?Locazione = Valore $0 < \text{Valore} < 255$
!Locazione = Valore $0 < \text{Valore} < 4294967295$
\$Locazione = Stringa

Il primo operatore '?' permette la modifica della Locazione nella quale viene memorizzato il Valore a otto bit. Il secondo opera in modo analogo, con la semplice differenza che il valore può essere a trentadue bit. In poche parole, se '?' altera una sola locazione, '!' ne altera quattro, con

notevole aumento del Valore assoluto memorizzabile

Per esempio:

?4000 = 255 POKa in 4000 il valore a otto bit 255
!4000 = 65535 POKa da 4000 a 4003 il valore 65535, conservando in 4001 il byte di ordine più elevato.

Per chi non lo ricordasse un Integer o valore intero a più di otto bit è sempre suddivisibile in più valori a otto bit. Così il valore 50000 può essere scritto come

$50000 = 80 + 256 * 195$

Ebbene, il byte di ordine più elevato, in questo caso, è 195. La formula generale per dividere un Numero a sedici bit in due da otto è:

Low = Numero MOD 256

High = Numero DIV 256

L'operatore \$ è quello senza dubbio più interessante, in quanto permette il diretto inserimento di una stringa in memoria. Così l'espressione:

\$4000 = "Evviva List!"

memorizza a partire dalla locazione 4000 la stringa "Evviva List!", ponendo automaticamente alla fine della stringa il codice 13 (fine stringa). Notate che oltre alla normale sintassi del tipo:

?(Locazione + Step) = Valore

dove Step è un valore che cambia rispetto agli altri (per es. in un ciclo di FOR), si può usare una espressione del tipo:

Locazione?Step = Valore

dove però la Locazione deve essere espressa per mezzo di una variabile. Per esempio:

INDIRIZZO = 300

INDIRIZZO?10 = 200

significa che nella Locazione $300 + 10$ deve essere memorizzato 200.

analogo discorso vale per l'operatore '!', mentre '\$' non permette questo particolare tipo di gestione.

Il modo per leggere una locazione ha una forma analoga al modo per scriverci dentro: infatti:

PRINT ?LOCAZIONE

PRINT !LOCAZIONE

PRINT \$LOCAZIONE

sono i tre modi di leggere un indirizzo permessi dal BASIC BBC. Ovviamente la prima espressione legge un valore a otto bit, la seconda uno a trentadue e la terza una stringa



di lunghezza indefinita, fino a che l'operatore non incontra un codice 13. Vi ricordiamo che in questo modo gli operatori diventano delle funzioni e pertanto godono di tutte le proprietà di queste ultime. Così, se battete:

\$4000="ho capito tutto!"

e poi:

PRINT \$4000

otterrete:

ho capito tutto!

Forme accettate sono anche:

PRINT Locazione?Step

PRINT Locazione!Step

dove Locazione è espressa tramite variabile BASIC (vedi prima).

Facile, no? Per concludere questo breve, ma utile articolo vi facciamo notare che il PC 128 S può manipolare numeri anche in formato esadecimale. Questo è scritto sul manuale e non c'è ancora molto da dire. Per esempio, se battiamo:

PRINT &FFFF;" "&FF;" "&BC

otteniamo:

65535 255 188

Ciò che non c'è scritto (o forse non ce ne siamo accorti) è come convertire un numero decimale in esadecimale. Anche in questo caso si usa un operatore, ed esattamente quello che sta sopra al simbolo di elevamento a potenza (ci scusiamo per non poterlo disegnare per motivi di stampa). Comunque, se non l'avete ancora trovato, è il simbolo a destra del simbolo di uguaglianza. Esso va usato alla stessa stregua del simbolo &, cioè davanti al numero decimale da convertire in esadecimale.

Termina qui questa breve carrellata su alcuni aspetti nascosti del PC 128 S. Vi assicuriamo comunque che c'è ancora molto da scoprire! Animo, dunque, e riacquistate quello spirito pionieristico a cui abbiamo accennato. Siamo ansiosi di vedere la redazione sommersa da vostre scoperte, vostri trucchi e trucchetti, i quali troveranno senza dubbio un posto sicuro tra le pagine della Vostra rivista.

RICEVIAMO E PUBBLICHIAMO...

Più veloce la scrittura in «inverse»

Riceviamo da un lettore di Ancona, il sig Walter Merli due routine interessanti a proposito di un listato già apparso su LIST di aprile, nella rubrica "Di tutto un po'", a pag. 58. Tale routine permetteva la scrittura in Inverse. Le routine inviate dal nostro lettore eseguono la medesima funzione, solo che "lo fanno più rapidamente, sfruttando un sottoprogramma in linguaggio macchina".

Routine 1

```
100 SCREEN 1:CLS:KEYOFF
110 GOSUB280
120 INPUT"COLORE SCRITTURA,SFONDO,BORDO"
;SC,SF,BD
125 COLORSC,SF,BD
130 A$="PROVA SCRITTURA INVERSE"
140 L=PEEK (VARPTR(A$))
150 IND=PEEK(VARPTR(A$)+2)*256+PEEK(VARPTR(A$)+1)
160 GOSUB240
170 B$"riprova scrittura inverse"
180 L=PEEK(VARPTR(B$))
190 IND=PK(VARPTR(B$)+2)*256+PEEK(VARPTR(B$)+1)
200 GOSUB240
210 PRINT A$
220 PRINTB$
230 END
240 FORK=0TOL
245 A=PEEK(IND+K)+96:IFA=126>0THEN270
250 POKE(IND+K),A
260 NEXTK
270 RETURN
280 IF PEEK(&HFC4F)=0THENI=2048ELSEI=0
290 FORK=32*8T0127*8
300 VPOKE(I+K+96*8),255-VPEEK(I+K)
310 NEXTK
320 RETURN
```

Alla riga 110 si va alla subroutine 280 che crea un alfabeto in Inverse: tutti i caratteri dal 32 al 126 vengono memorizzati in Inverse a partire dal carattere 127 in poi.

Alla riga 140 viene richiamata dalla VLT la lunghezza della stringa da convertire.

Alla riga 150 viene memorizzato in IND l'indirizzo di memorizzazione della stringa.

Così anche alle righe 180 + 190.

Viene poi richiamata la subroutine 240 che provvede a sostituire ogni carattere della stringa con il corrispondente carattere in inverse.

Routine 2

Il funzionamento è lo stesso della routine 1 solo che le subroutine 280 e 240, gli indirizzi della VLT, sono gestiti da L/M.

```
100 CLEAR200,&HEA5F
110 AT=&HEA60:DEFUSR=A!
120 FORI=0T076
130 READB:POKEA!+I,B
140 NEXTI
150 DEFUSR1=&HEA92
160 SCREEN1:CLS:KEYOFF
170 U=USR(0)
180 INPUT"COLORE SCRITTURA,SFOND96;Üj8Ü
dgjhüSF,BD
190 A$LINE"ÄÏ»ÖiCiüi-888iic-«ÖNiiN":U$=
USR1(A$)
195 COLORSC,SF,BD
200 LOCATE,12:PRINTA$
210 B$"riprova scrittura inverse":U$=U
SR1(B$)
220 PRINTB$
230 END
1000 DATA58,175,252,254,2,202,202,71,25
4,1,202,115,234,33,0,9,195,118,234
1010 DATA33,0,1,1,0,3,17,0,3,205,74,0,2
13,95,62,255,147,209,25,205,77,0
1020 DATA237,82,35,11,120,177,32,235,20
1,254,3,194,202,71,235,126,254,0
1030 DATA 200,71,35,94,35,86,26,198,96,
254,127,56,4,18,19,16,245,201
```


C-64

+ Disk Drive 1571

ATTENTI ALLA TESTINA

Quando caricate del software di tipo commerciale, che dovrete aver regolarmente acquistato presso qualche rivenditore autorizzato e non presso qualche vostro amico copiatore, il drive 1541 spesso emette un rumore forte e fastidioso.

Questo tipo di inconveniente è causato dalla protezione anti-copia installata dall'autore.

A lungo andare, tale inconveniente può causare il disallineamento della testina di lettura/scrittura del vostro drive 1571.

Se provate a digitare, in modo diretto, la seguente linea di istruzioni prima di caricare il software, il pericolo di disallineamento dovrebbe essere scongiurato.

Questo «trucchetto» funziona nel 90% dei casi, ma nel malaugurato caso che non funzionasse esso non interferisce in alcun modo sul corretto caricamento e funzionamento del programma.

```
1 OPEN 1,8,15:PRINT # 1,"M-W"CHAS$(106)CHAS$(0)CHAS$(1)CHRS$(13):CLOSE 1
```

C-64

+ Disk Drive 1571

FILES UTILITY

Questa routine può essere usata, sul vostro C64 per controllare se un determinato file è presente sul disco e per verificare se un floppy è dotato di protezione contro la scrittura (WRITE PROTECTION).

È infatti ottima consuetudine, prima di manipolare dei files fare dei controlli molto accurati.

Se il file che state cercando sul vostro floppy non esiste, la linea 130 genererà un errore il cui numero di codice è 62: tale errore equivale al messaggio «FILE NOT FOUND» che vi informa sul fatto che il file che state ricercando non è presente su quel disco. Se non vi serve fare controlli sulla WRITE-PROTECTION, sostituite nel primo statement OPEN la lettera A con la lettera R.

```
100 REM *****
101 REM * FILE UTILITY *
102 REM *****
103 INPUT "Nome del file da cercare ";f$
110 OPEN 2,8,2,F$+"",F,A":CLOSE 2
120 OPEN 15,8,15:INPUT #15,A,B$:CLOSE 15
130 IFA>1THENPRINTA,B$:END
140 PRINT"* IL FILE E' SUL DISCO * "
```

C-128

TEST MODO 64/128

La seguente istruzione permette all'utente di conoscere se il programma attivo sul proprio computer funziona in modo 128 o 64.

La conoscenza di tale informazione è possibile grazie ad un trucchetto: le istruzioni FRE(0) e FRE(1) nel modo 64 restituiscono due valori distinti che rappresentano rispettivamente la quantità di bytes liberi per il programma e per le variabili.

Dopo l'esecuzione di tale istruzione, la variabile M conterrà il valore 128 o il valore 64 a seconda del modo nel quale state lavorando.

```
100 M = 128:IF FRE(0) = FRE(1) THEN M = 64
```

C-64

TASTO RESET

Uno dei difetti principali del pur valido C-64 è quello di non avere a disposizione il tasto di reset attraverso il quale è possibile riportare il vostro computer allo stato iniziale.

Per ovviare a tale inconveniente basterà includere nella vostra biblioteca software questa semplice utility, che una volta «lanciata» trasformerà il tasto RESTORE del vostro C-64 in un efficientissimo tasto di RESET.

```
10 FORA = 4070TO40706:READB:POKEA,B:NEXT:FORA = 32770TO32776:READB:POKETA,B:NEXT:NEW
20 DATA 32,248,252,0,159,195,194,205,56,48
```

C-128

INPUT PROMPT DEL VOSTRO C128

Per eliminare il punto interrogativo dall'istruzione di input digitate:

```
POKE 21,64
```

Subito dopo lo statement di input digitate

```
POKE 21,0
```

per riportare il sistema alla normalità.

ATTENZIONE: se non digitate l'istruzione contenente il secondo POKE il vostro computer non funzionerà a dovere.

MSX

ISTRUZIONI... AL MICROSCOPIO

IL COMANDO RENUM NEGLI MSX

Nessuno ne parla mai perché pare un comando scontato e senza grosso peso specifico ai fini di un qualsiasi listato, che, alle volte, invece, possono letteralmente salvare il contenuto di un programma in specie se questo viene digitato senza un relativo flow-chart che ne codifichi almeno sommariamente inizio e fine.

Cosicché, alla conclusione di questa pagina, sarete sorpresi delle capacità del vostro RENUM e il vostro bagaglio di programmatore ne trarrà giovamento, non contando che un piccolo segreto condirà il finale dell'articolo.

A cosa serve

Il comando RENUM serve a rinumerare le linee di programma, ma non solo dalla prima riga e neanche solo al passo di dieci.

La sintassi

RENUM nuovo numero riga, vecchio numero riga, incremento, dove:

NUOVO NUMERO DI RIGA = al primo numero di linea che dovrà essere usato nella sequenza (per default 10)

VECCHIO NUMERO DI RIGA = alla linea da cui dovrà partire la rinumerazione (per default la prima)

INCREMENTO = al passo numerico di ogni linea (per default 10)

Esempi pratici

Poniamo a base degli esempi uno pseudo-listato di partenza e vediamo come reagisce ai vari RENUM.

```
10 REM A
20 REM B
30 REM C
40 GOTO 20
```

Quindi diamo il RENUM 40 e avremo:

```
40 REM A
50 REM B
60 REM C
70 GOTO 50
```

Da notare come il GOTO di linea 70 si riporti sempre al secondo numero di linea.

Ancora al list di partenza diamo il RENUM 52,,1 e avremo:

```
52 REM A
53 REM B
54 REM C
55 GOTO 53
```

Con il RENUM 80,30,5 avremo invece:

```
10 REM A
20 REM B
80 REM C
85 GOTO 20
```



Come vedete è una variazione utilissima per lasciare invariata la numerazione di una prima parte del listato, rinumerandone a piacere la seconda. ancora con RENUM „8 avremo:

```
10 REM A
18 REM B
20 REM C
34 GOTO 18
```

Da notare come le virgole rimpiazzino la sintassi del vecchio numero di riga.

Il RENUM nel listato !

Non l'avevo accennato prima per ottenere un effetto sorpresa, poiché certamente molti di voi non sanno che il RENUM è anche una particolare istruzione.

Così, ora, provate a digitare questo list:

```
10 REM A
20 REMB
30 RENUM 40
40 GOTO 20
```

Quindi date il RUN e avrete un OK dal computer, allora digitate LIST e apparirà:

```
40 REM A
50 REM B
60 RENUM 70
70 GOTO 50
```

Sorpresi, vero? Ebbene, sì, il RENUM rinumerava anche sé stesso, portandosi in questo caso alla quota in cifre dell'ultimo numero di linea rinumerato.

Ma non è finita, e mantenendo fisso l'ultimo list d'esempio, digitiamo ancora il RUN, aspettiamo l'OK e richiediamo nuovamente il LIST.

In questo modo avremo:

```
70 REM A
80 REM B
90 RENUM 100
100 GOTO 80
```

Come vedete l'azione del RENUM continua e si ripeterà ad ogni RUN.

Naturalmente qualsiasi variante RENUM già vista nei precedenti esempi potrà essere inclusa anche in questo tipo di listati, consentendo al programmatore esperto di strutturarsi adeguatamente e senza perdite di tempo un listato con rinumerazione ad hoc.

Non avevo ragione nel dire che avremmo imparato qualcosa di utile?

Sfruttare al meglio un computer vuol dire conoscerlo a fondo, anche nei più remoti particolari e LIST è qui per questo.

O. Contenti

ZX SPECTRUM**Come usare alcune interessanti routines della ROM**

Per programmare in Linguaggio Macchina (per gli amici L/M) occorre conoscere il Linguaggio Macchina e su questo c'è poco da discutere. Sembra anche abbastanza ovvio che per scrivere programmi interessanti non solo bisogna conoscere il linguaggio in questione, ma occorre conoscerlo anche molto bene. Precisiamo che ancora ciò non basta. Per scrivere programmi di un certo livello è pressoché indispensabile conoscere ANCHE la propria macchina. Le ragioni di ciò sono molto chiare: se si vuole scrivere un programma grafico in L/M occorrerà usare alcune routines che disegnano cerchi, linee e punti. Perché allora riscriverle daccapo se queste sono tranquillamente residenti nella ROM? Ancora: siamo interessati alla stesura di uno studio di funzione in L/M: qui, oltre alle routines grafiche, bisogna lavorare con funzioni, con numeri, quasi sempre del tipo «rognoso numero a virgola mobile». Chi sarebbe masochista al punto di ricreare l'intera gestione dei numeri in Floating Point, considerando il fatto che detta gestione esiste già immortalata sulla ROM? Per questi motivi ci sembra a dir poco doveroso andare a ficcanasare nella ROM per riuscire ad identificare e a comprendere il funzionamento di alcune subroutines, specialmente quelle relative alla grafica: con il valido aiuto di una buona documentazione e sorretti da ferrea volontà siamo riusciti a isolare alcuni indirizzi molto interessanti. Per una facile e rapida consultazione, useremo il seguente metodo espositivo:

- 1) Nome simbolico della routine e suo indirizzo decimale ed esadecimale.
- 2) Descrizione del suo funzionamento.
- 3) Qualche esempio (in ASSEMBLER e codici decimali; questi ultimi possono essere direttamente inseriti a partire da un indirizzo qualsiasi e chiamati con una regolare USR alla locazione di partenza scelta).

CHANOPEN 5633 dec. 1601 esa.

Questa subroutine apre il canale il cui numero è contenuto nell'accumulatore. I canali più interessanti sono il video, con il codice 2 e la stampante, con codice 3. Si noti che durante l'esecuzione di un programma prima aprire il canale 2 e poi inviare i caratteri da stampare. In altre parole, il video va trattato alla stessa stregua degli altri canali e quindi va attivato con una regolare chiamata.

ES.
 62, '2' LD A,2 ; codice del canale video
 205,1,23 CALL 5633 ; apri il canale specificato
 201 RET ; ritorna al sistema

(N.B. - Il 2 tra gli apici indica che tale valore non è in codice L/M ma un dato che può essere cambiato).

A scopo chiarificatore indichiamo come vanno usati i codici decimali. Scegliamo come locazione di partenza, ad esempio, l'indirizzo 50000. allora in tale indirizzo memorizzeremo con delle POKES o con un programma specifico (come il nostro HEX LOADER pubblicato ad APRILE) il codice 62. Nella locazione seguente, in questo caso in 50001, il codice 2, passibile di modifica, ecc. Per eseguire il programma basta digitare RANDOMIZE USR 50000.

PLOT 8927 dec. 22DF esa.

Setta a 1 il PIXEL le cui coordinate sono contenute nei registri B e C. Le ascisse, cioè X, sono conservate nel registro C mentre le coordinate, Y, in B.

ES.
 14, '100' LD C,100 ; C contiene X
 6, '110' LD B,110 ; B contiene Y
 205,223,34 CALL 8927 ; PLOT X,Y
 201 RET ; finito

DRAW 9402 DEC. 24BA esa.

Traccia una linea dall'ultimo punto disegnato con le stesse modalità del comando BASIC DRAW. C contiene il numero di PIXELs lungo le ordinate mentre B secondo le ascisse. Inoltre i registri E e D devono contenere, rispettivamente, il segno di C e B: 1 significa positivo, 255 negativo. Il valore positivo o negativo assume lo stesso significato della direzione di tracciamento del BASIC.

ES.
 6, '110' LD C,110 ; C = ABS X
 14, '100' LD B,100 ; B = ABS Y
 22, '1' LD E,1 ; E = SGN X
 30, '1' LD D,1 ; D = SGN Y
 205,186,36 CALL 9402 ; DRAW X,Y considerando il segno
 217 EXX ; questa sequenza di istruzioni serve per
 33,88,39 LD HL,10072 ; ridefinire un certo indirizzo conservato
 217 EXX ; nel set alternativo in HL.
 ; Ciò viene fatto per evitare
 ; dei ritorni poco simpatici al sistema
 201 RET ; finito

CIRCLE 9005 dec. 232D esa.

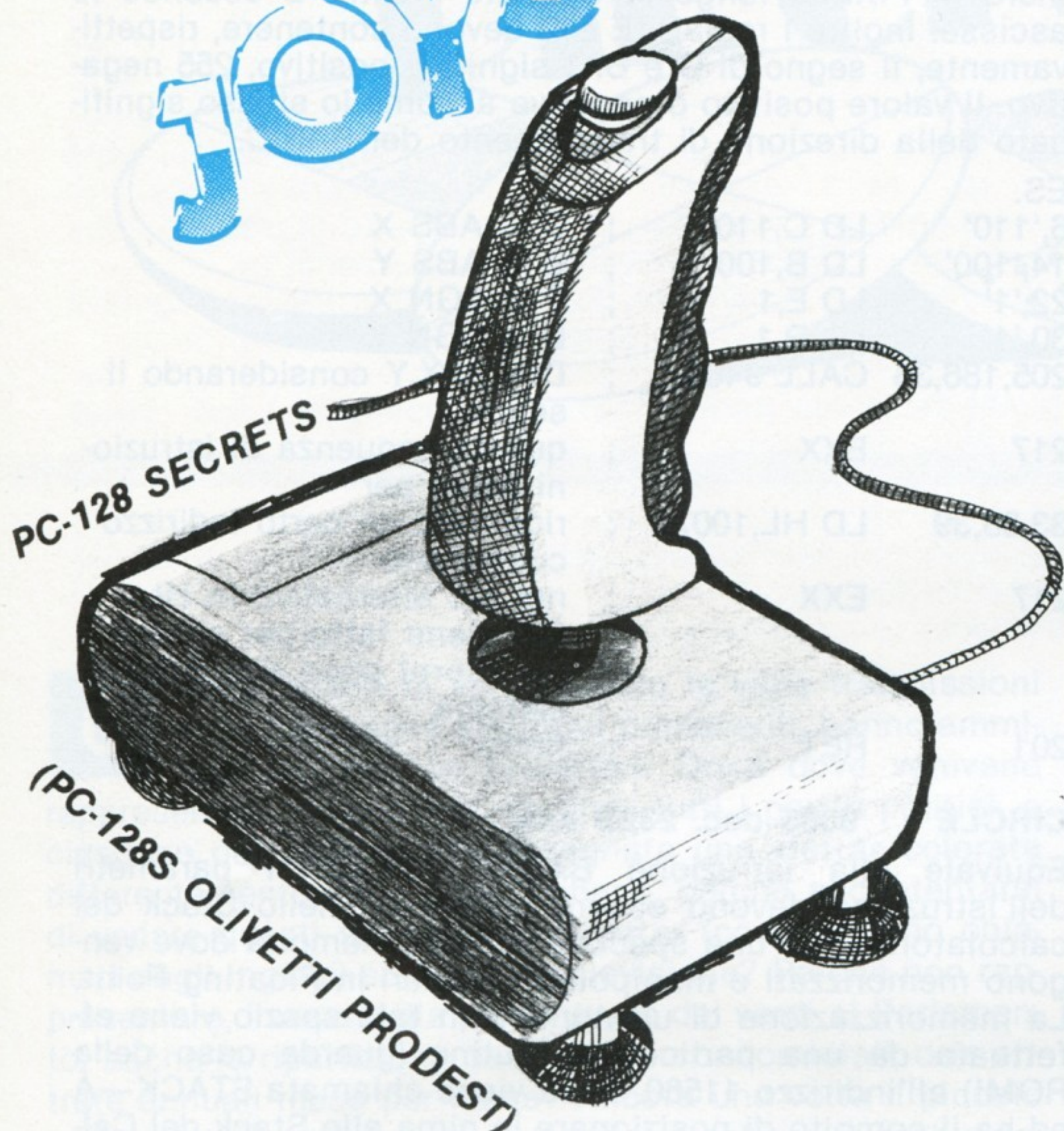
Equivalenza alla istruzione BASIC CIRCLE. I parametri dell'istruzione devono essere conservati nello Stack del calcolatore, cioè una speciale zona di memoria dove vengono memorizzati e manipolati i numeri in Floating Point. La memorizzazione di un numero in tale spazio viene effettuata da una particolare routine (guarda caso della ROM!) all'indirizzo 11560. Essa viene chiamata STACK—A ed ha il compito di posizionare in cima allo Stack del Calcolatore il valore conservato in A.

ES.
 62, '100' LD A,100 ; carica A, in sequenza, con
 ; 100, 100
 205,40,45 CALL 11560 ; e 40, che rappresentano X,Y
 ; e Radius
 62, '100' LD A,100 ; Memorizza i valori sullo
 ; Stack del
 205,40,45 CALL 11560 ; Calcolatore chiamato
 ; STACK—A al-
 62, '40' LD A,40 ; l'indirizzo 11560. Attenzio-
 ; ne alla se-
 205,40,45 CALL 11560 ; quenza di salvataggio dei
 ; dati: prima X, poi Y e quindi
 ; il raggio
 205,45,35 CALL 9005 ; CIRCLE X,Y,Radius
 217 EXX ; come per DRAW, ripristina
 ; un certo valore
 33,88,39 LD HL,10072 ; 'un felice ritorno' al BASIC
 217 EXX ;
 201 RET ; ritorna al BASIC

Termina qui questa breve ma interessante (e soprattutto utile) trattazione di alcune ROM-subroutines. Siamo sicuri che molti dei nostri affezionati lettori sincleristi stanno già facendo lavorare il loro cervello al massimo programma in BASIC in un suo «fratello» in L/M. Noi di LIST siamo molto curiosi di conoscere il frutto del nostro ma soprattutto del vostro lavoro personale. Rinnoviamo quindi l'invito all'invio di programmi originali ed interessanti elaborati magari alla luce di queste nuove conoscenze.

per saperne di più...¹K

IL JOYSTICK



L'argomento di questo numero tratta un indispensabile accessorio del computer, sulla gestione del quale il manuale è piuttosto avaro di particolari: il joystick.

■ Come è noto, il joystick, è essenzialmente costituito da 5 interruttori, 4 dei quali controllati dalla posizione della leva (alto, basso, destra, sinistra) ed uno dal pulsante di fuoco.

Ciascuna di queste posizioni attiva separatamente uno ed un solo interruttore: sono tuttavia possibili situazioni più complesse, quali le posizioni intermedie della leva (ad esempio tra alto e destra, tra basso e sinistra, ecc.) che attivano 2 interruttori contemporaneamente; se, insieme allo spostamento della leva, viene premuto anche il pulsante di fuoco, possono risultare attivati fino a 3 interruttori nello stesso momento.

Il computer deve essere in grado di «riconoscere» tutte queste combinazioni in modo da poter scegliere, secondo le istruzioni del programma, l'ideale procedura da seguire.

■ Nel BBC BASIC le informazioni sullo «status» del joystick (cioè il riconoscimento di quali interruttori sono, ad un certo istante, chiusi o aperti) risiedono nelle funzioni ADVAL(0), ADVAL(1) e ADVAL(2) che assumono valori numerici diversi in funzione dello status stesso: tali cifre possono essere visualizzate mediante l'istruzione "PRINT ADVAL(0), ADVAL(1), ADVAL(2)".

■ Le funzioni ADVAL(1) e ADVAL(2) assumono valori numerici variabili da 0 a 65535: muovendo la leva del joystick verso sinistra, ADVAL(1) cresce fino a raggiungere 65535; ulteriori successivi spostamenti nella stessa direzione non provocano modifiche di tale valore. Muovendo la leva verso destra ADVAL(1) decresce fino a stabilirsi su zero. Analogo comportamento presenta la funzione ADVAL(2) nei riguardi degli spostamenti in senso verticale: muovendo la leva verso l'alto ADVAL(2) cresce mentre verso il basso diminuisce fino agli stessi valori limite. Quanto esposto può essere facilmente verificato mediante le seguenti righe di programma:

```
> 10 MODE 128
> 20 PRINT ADVAL(1),ADVAL(2):GOTO 20
```

che visualizzano due colonne di numeri corrispondenti, rispettivamente, ai valori delle due funzioni; muovendo il joystick se ne possono agevolmente constatare le variazioni.

■ La funzione ADVAL(0) è molto più complessa poiché in essa risiede l'effettivo controllo dello status del joystick: Battete, ad esempio, il seguente programma:

```
> 10 MODE 128
> 20 PRINT ADVAL(0):GOTO 20
```

Osserverete, qualunque sia la posizione del joystick, una colonna di numeri in una tale continua e rapida modifica da renderne impossibile la lettura; la situazione migliora notevolmente inserendo un punto e virgola nella riga 20:

```
> 20 PRINT ADVAL(0);:GOTO 20
```

Le cifre sono ora visualizzate in righe di 10 per volta ed è possibile leggerne il valore. Tentate una sistematica analisi dei valori della funzione muovendo la leva del joystick in tutte le direzioni e premendo alternativamente il pulsante di fuoco: constaterete che le cifre visualizzate, pur essendo variabili, non subiscono il costante incremento o decremento presentato da ADVAL(1) e ADVAL(2) e che, a parità di posizione della leva e del pulsante di fuoco, i numeri sono sempre i medesimi. Confrontate i numeri letti con la seguente tabella:

VALORE DI ADVAL(0) CON PULSANTE DI FUOCO

Posizione di leva	Rilasciato	Premuto
1 Centro	256-512	257-513
2 Sinistra	272-528	273-529
3 Basso	288-544	289-545
4 Basso-sinistra	304-560	305-561
5 Alto	320-576	321-577
6 Alto-sinistra	336-592	337-593
7 Destra	384-640	385-641
8 Basso-destra	416-672	417-673
9 Alto-destra	448-704	449-745

Analizzando le cifre riportate nella tabella, si possono trarre alcune interessanti conclusioni:

1. - Ad ogni posizione della leva corrisponde una coppia di numeri la cui differenza è costantemente pari a 256; la coppia di valori più bassi corrisponde alla posizione centrale della leva (nessun interruttore chiuso).

2. - Fino alla posizione n. 6, la differenza tra termini omologhi di due coppie successive è pari a 16; la differenza

dei valori tra la posizione 6 e 7 è 48, mentre sia tra 7 e 8 che tra 8 e 9 è pari a 32.

3. - La pressione del pulsante fuoco provoca l'incremento di una unità di tutti i valori assunti dalla funzione ADVAL(0); è interessante notare che, al di là del puro incremento numerico, il pulsante di fuoco rende *dispari* tutti i valori della funzione.

Da questa analisi possono essere tratte alcune conclusioni molto utili in fase di programmazione:

1. - Poiché la differenza tra i termini di una coppia è costante, non è necessario tener conto di entrambi i numeri; è sufficiente assegnare la funzione ADVAL(0) ad una qualsiasi variabile, controllarne il valore e sottrarre 256 se esso supera il numero 448: si ottiene in tal modo una variabile in cui è sempre contenuto il termine minore della coppia.

>....Z=ADVAL(0):IF Z>448 THEN Z=Z-256

2. - Tutti i valori ADVAL(0) (quando il pulsante di fuoco è rilasciato) sono divisibili per 16; pertanto, dividendo la funzione per 16 e sottraendo 15, si ottengono dei numeri interi, di piccolo valore, a partire dal n. 1 corrispondente quest'ultimo alla posizione di riposo del joystick (ADVAL(0)=256 — 256/16-15=1). Il calcolo può essere eseguito contemporaneamente al controllo descritto al precedente punto 1, risparmiando così una riga di programma:

>....Z=ADVAL(0):IF Z>448 THEN Z=(Z-256)/16-15 ELSE Z=Z/16-15

I valori di Z così ottenuti possono essere agevolmente utilizzati in una istruzione ON Z PROC.... oppure ON Z GOSUB....

3. - Il controllo della pressione del pulsante di fuoco può essere facilmente ottenuto verificando se il valore di ADVAL(0) è **pari** (INT(Z/2)=Z/2) o **dispari**(INT(Z/2)<>Z/2).

■ Le considerazioni sopra esposte sono state utilizzate per il piccolo programma che segue, il quale gestisce il movimento di una pallina sullo schermo mediante il joystick; sono inoltre visualizzate, nell'ultima riga, le coordinate istantanee della pallina, il valore della funzione ADVAL(0) e quello della variabile Z che indirizza le procedure in base alla posizione della leva del joystick.

■ Molte istruzioni hanno una funzione puramente «decorativa», ma quelle che interessano il movimento possono essere così riassunte:

— alla **R.40** viene inserito il valore istantaneo di ADVAL(0) nella variabile Z e viene controllato se quest'ultima è pari o dispari: nel primo caso il programma prosegue, altrimenti è attivata la procedura di «sparo» che disegna cerchi concentrici intorno alla pallina.

— la **R.50** si incarica di controllare il valore di Z e di ridurlo a multipli di 1.

— alla **R.70** viene scelta la procedura da seguire in relazione al valore di Z e, quindi, alla posizione della leva del joystick.

— le procedure da **R.90 a R.280** si incaricano di incrementare o decrementare opportunamente le coordinate di stampa della pallina in relazione alla direzione scelta con il joystick; la PROC A (R.290) esegue un ciclo a vuoto e viene utilizzata sia quando la leva è in posizione centrale

>LIST

```
10REM*** JOY DEMO ***
20MODE129:VDU23;8202;0;0;0;23,250,60,
126,126,255,255,126,126,60:X=19:Y=15
30PRINTTAB(0,30)"
":PRINTTAB(0,30)"X=";X;"
Y=";Y:PRINTTAB(X,Y)CHR$250:SOUND1,-10
,X*6,1:SOUND2,-10,Y*9,1
40Z=ADVAL(0):PRINTTAB(15,30);Z:IFINT
(Z/2)<>Z/2PROCsparo:COLOURRND(3):GOTO
30
50IFZ>450Z=(Z-256)/16-15ELSEZ=Z/1
6-15
60PRINTTAB(30,30);Z:PRINTTAB(X,Y)CH
R$32
70ONZPROCA,PROCsinistra,PROCbasso,P
ROCbasso_sinistra,PROCalto,PROCalto_sini
stra,PROCA,PROCA,PROCdestra,PROCA,PROCba
sso_destra,PROCA,PROCalto_destra
80GOTO30
90DEFPROCsinistra:X=X-1:IFX<0X=0
100ENDPROC
110DEFPROCdestra:X=X+1:IFX>39X=39
120ENDPROC
130DEFPROCalto:Y=Y-1:IFY<0Y=0
140ENDPROC
150DEFPROCbasso:Y=Y+1:IFY>28Y=28
160ENDPROC
170DEFPROCalto_destra:X=X+1:IFX>39X=
39:GOTO190
180Y=Y-1:IFY<0Y=0:X=X-1
190ENDPROC
200DEFPROCalto_sinistra:X=X-1:IFX<0X
=0:GOTO220
210Y=Y-1:IFY<0Y=0:X=X+1
220ENDPROC
230DEFPROCbasso_destra:X=X+1:IFX>39X
=39:GOTO250
240Y=Y+1:IFY>28Y=28:X=X-1
250ENDPROC
260DEFPROCbasso_sinistra:X=X-1:IFX<0
=0:GOTO280
270Y=Y+1:IFY>28Y=28:X=X+1
280ENDPROC
290DEFPROCA:ENDPROC
300DEFPROCsparo:SOUND0,-10,20,10:X1=X*
2+16:Y1=Y*32+16:FORK=5TO15:GCOLRND(
5)-1,RND(3):MOVEX1,1023-Y1:PLOT149,X1+K
2,1023-Y1-K^2:NEXT:CLG:ENDPROC
```

(nessuno spostamento) che come «riempitivo» nell'istruzione ON Z PROC... per i valori che Z non può mai assumere, corrispondenti agli incrementi tra due coppie contigue superiori a 16.

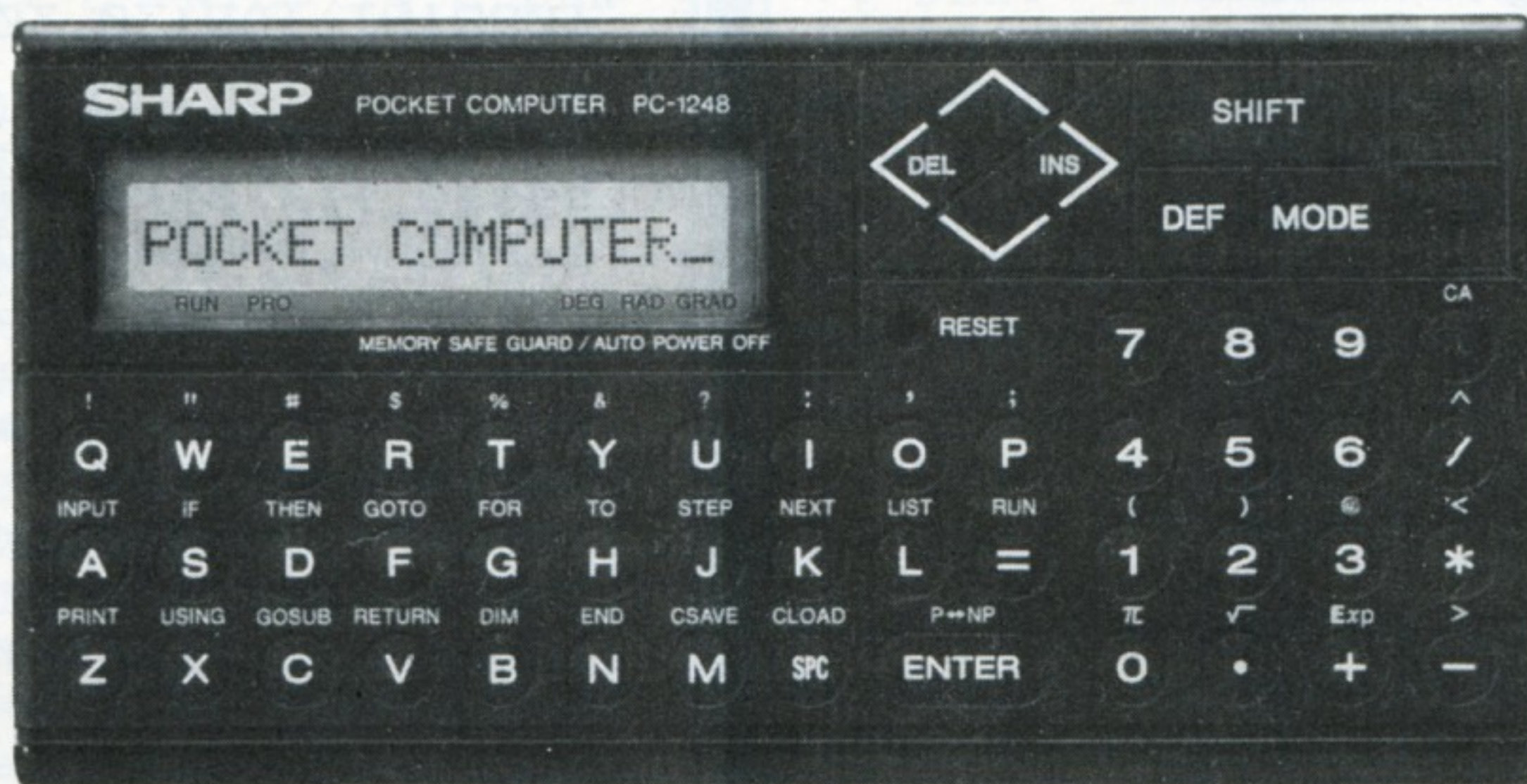
■ Come potete notare, il movimento della pallina è già piuttosto veloce; tuttavia esso è rallentato a causa del tempo necessario per l'esecuzione di alcune istruzioni accessorie. Sostituendo alcune righe con le seguenti, il programma si riduce allo stretto indispensabile per la gestione del movimento:

```
>30PRINTTAB(X,Y)CHR$250
>40Z=ADVAL(0):IFINT(Z/2)<>Z/2PROCsparo:GOTO30
>PRINTTAB(X,Y)CHR$32
```

Osservate ora la velocità di spostamento e sbalordite. È quasi un... «linguaggio macchina»!

8 Kb RAM

**Continua la
nostra rassegna
dei pocket
computers più
diffusi sul mercato.**



**Questo mese la
nostra attenzione
è caduta sul
piccolissimo ma
potente PC-1248**

Trasformazione in senso funzionale del già noto PC-1247, lo SHARP PC-1248 si caratterizza per la sua originale estetica. contrariamente ai pocket più diffusi è infatti totalmente nero, con i tasti «soft touch» rotondi e ricoperti da una sottile membrana a tenuta che protegge la macchina da polvere e umidità.

Piccolo come un pacchetto di sigarette e leggerissimo (pesa poco meno di un etto), il PC-1248 consta di una tastiera alfabetica con 18 tasti predefiniti con istruzioni Basic (utilissimi per sveltire le digitazioni in fase di programmazione e per evitare noiosi errori), un tastierino numerico, ed alcuni tasti separati per lo scorrimento delle righe dei programmi e la gestione del cursore sul display. Quest'ultimo è a cristalli liquidi, formato da 16 caratteri a matrice di punti 5x7.

Pur essendo molto piccolo e maneggevole, il PC-1248 è dotato di una memoria molto potente (quasi 8 Kb RAM permanente), cosa che lo rende assai utile anche a chi abbia l'abitudine di redigere programmi lunghi e complessi. Il Basic di questo pocket è abbastanza completo, in quanto presenta tutte le istruzioni standard del linguaggio, ed anche qualcuna un po' particolare e rara per un computer così piccolo, come ON... GOTO e ON... GOSUB, TRON e TROFF (per il debug in fase di programmazione), AREAD (istruzione che vi consente, eseguendo programmi richiamabili con il tasto DEF, di fare assumere alla variabile specificata il valore di quanto presente sullo schermo).

Il Basic consente inoltre di usare variabili alfanumeriche ad una lettera, ed anche matrici bidimensionali.

Come in tutti i pocket della SHARP, ANCHE IL PC-1248 è dotato di numerose funzioni matematiche (trigonometriche e trigonometriche inverse, esponenziali e logaritmiche), e inoltre riserva istruzioni specifiche (DEGREE, RADIANT, GRAD) per le conversioni angolari.

Alcune considerazioni meritano poi alcune funzioni speciali di questo computer da taschino: il comando per i suoni BEEP, l'autospegnimento dopo qualche minuto di non uso (per evitare un inutile dispendio di energia), nel PASSWORD definibile per proteggere il contenuto della memoria da operatori indesiderati.

Per quanto riguarda la possibilità di collegamenti periferici, il PC-1248 non è certo inferiore agli altri esemplari che offre il crescente mercato. La sua porta a 11 PIN (disposta su uno dei due lati più corti) offre la possibilità di collegarsi con la SHARP CE-125, stampante termica da 24 caratteri per riga con registratore a micro cassette, o con la SHARP CE-126 P (che è la stessa stampante del modulo sopra descritto, solo che non comprende il registratore ma il cavo per collegarlo). Altrimenti, per chi volesse collegare questo pocket al solo registratore senza avere bisogno della stampante, è sufficiente l'interfaccia SHARP CE-124.

L'ultima notazione è per la durata delle batterie di questo pocket: la casa garantisce una autonomia di 120 ore di utilizzo medio (questo prevede un rapporto di 1 a 5 tra il tempo di programmazione e la durata di accensione del display), quanto basta per fare di questo piccolo e scuro computer un vademecum insostituibile.

COMPUTER TASCABILE SHARP PC-1248

CARATTERISTICHE TECNICHE

MODELLO	SHARP PC-1248
DIMENSIONI	135 x 70 x 11 mm.
PESO	85 gr.
TASTIERA	53 TASTI.
VISORE	A CRISTALLI LIQUIDI. 16 CARATTERI.
MEMORIA RAM	7,422 Kb.
TEMPERATURA DI FUNZIONAMENTO	0 °C-40 °C.
CONSUMO	0,05 W.
ALIMENTAZIONE	2 PILE AL LITIO CR 2032.

CALCOLO COMBINATORIO

Immaginate di avere dimenticato la combinazione della vostra cassaforte e di volerla trovare per tentativi. A seconda delle cifre che compongono la vostra combinazione e dell'arco in cui esse possono variare (0-9, 0-99, ecc.) il calcolo combinatorio vi permetterà di conoscere con esattezza la quantità di tutte le possibili disposizioni delle cifre in questione.

I concetti base del calcolo combinatorio sono due: gli oggetti e la classe. Il numero degli OGGETTI è dato dalla quantità di elementi di qualsiasi natura (siano essi cifre di una combinazione, simboli di una slot-machine o del totocalcio, e così via) su cui opera questo tipo di calcolo. Per CLASSE si intende invece il numero di posizioni che possono essere assunte dagli oggetti (per intenderci: nelle slot-machine più diffuse la classe sarà 3, nelle schedine del totocalcio sarà 13, e così via).

Il nostro programma è diviso in 5 parti (ciascuna richiamabile con un tasto predefinito), a seconda del tipo di calcolo che vogliamo svolgere.

Supponiamo di avere 3 oggetti (es.: i numeri 1 2 e 3); vogliamo considerare tutti i gruppi possibili formati da questi oggetti. Essi saranno: 12, 13, 21, 23, 31, 32; abbiamo così le DISPOSIZIONI di classe 2, di 3 oggetti. Se escludiamo da queste quelle in cui si ripetono le stesse cifre (eliminiamo cioè 21 in quanto già le due cifre sono presenti in 12, e così via), avremo le COMBINAZIONI: 12, 13, 23. Se vogliamo poi tenere conto dei gruppi che si ottengono ripetendo più volte la stessa cifra (11, 22, 33), parleremo allora, rispettivamente, di DISPOSIZIONI CON RIPETIZIONI (11, 12, 13, 21, 22, 23, 31, 32, 33), e di COMBINAZIONI CON RIPETIZIONI (11, 12, 13, 22, 23, 33).

Il quinto tipo di calcolo combinatorio si presenta quando il numero degli oggetti è uguale alla classe; in questo caso parleremo di PERMUTAZIONI di N oggetti. Ad esempio, le permutazioni dei tre soggetti presi in considerazione negli esempi precedenti sono: 123, 132, 213, 231, 312, 321. Come già accennato abbiamo diviso in 5 parti il nostro programma, ognuna corrispondente ad un tipo di calcolo e richiamabile con i tasti definibili, secondo la tabella che segue.

DEF A	: DISPOSIZIONI INPUT: OGGETTI E CLASSE
DEF B	: DISPOSIZIONI CON RIPETIZIONI INPUT: OGGETTI E CLASSE
DEF C	: COMBINAZIONI INPUT: OGGETTI E CLASSE
DEF D	: COMBINAZIONI CON RIPETIZIONI INPUT: OGGETTI E CLASSE
DEF F	: PERMUTAZIONI INPUT: OGGETTI

Per quanto riguarda gli INPUT dei programmi, vi segnaliamo che oggetti e classi non possono essere minori di 1, e che nelle disposizioni e combinazioni senza ripetizioni la classe non deve essere maggiore degli oggetti. Ogni tipo di errore di immissione dei dati sarà comunque controllato dal computer, che vi permetterà la ripetizione dell'input. Con questo programma potrete soddisfare anche piccole curiosità. Per esempio: vi siete mai chiesti quante colonne dovrete giocare al totocalcio per avere la certezza matematica di fare 13? Sono esattamente 1.594.323, per la modica (?) cifra di 797.161.500 di lire. Abbiamo ottenuto questo risultato richiamando con DEF B il nostro programma, e immettendo, alle richieste di input del computer, il numero d oggetti (3) e la classe (13).

Un altro esempio di utilità pratica: volete organizzare un torneo di calcio con 9 squadre, da disputarsi a girone unico con partite di andata e ritorno. Ebbene sarà sufficiente richiamare il programma con DEF A ed immettere gli oggetti (9) e la classe (2), per ottenere il risultato, che in questo caso sarà di 72 partite.

Il programma «gira» su tutti i poket della SHARP che hanno i tasti definibili (su quasi tutti, quindi). Per gli altri, comunque, basterà fare partire il programma, con il RUN, ma con un'istruzione di GOTO, seguita dal numero di linea corrispondente all'inizio della parte di programma prescelta.

```
1 REM *****
2 REM *   CALCOLO COMBINATORIO   *
3 REM * DI M. MARZANO E M. ERBANI *
4 REM *           @ 1987           *
5 REM *****
10 "A":PRINT " * DISPOSIZIONI * "
20 INPUT "NUM. OGGETTI ?";N:IF N<1 THEN
  GOSUB 1000:GOTO 20
30 INPUT "CLASSE ?";K:IF K<1 THEN GOSUB
  1000:GOTO 30
40 IF K>N THEN GOSUB 1000:GOTO 20
50 R=1:FOR I=N TO (N-K+1) STEP -1
60 R=R*I:NEXT I
70 PRINT"NUMERO DISPOS. =":PRINT R:END
100 "B":PRINT " * DISP. CON RIP. * "
110 INPUT "NUM. OGGETTI ?";N:IF N<1 THEN
  GOSUB 1000:GOTO 110
120 INPUT "CLASSE ?";K:IF K<1 THEN GOSUB
  1000:GOTO 120
130 R=N*K:GOTO 70
200 "C":PRINT " * COMBINAZIONI * "
210 INPUT "NUM. OGGETTI ?";N:IF N<1 THEN
  GOSUB 1000:GOTO 210
220 INPUT "CLASSE ?";K:IF K<1 THEN GOSUB
  1000:GOTO 220
230 IF K>N THEN GOSUB 1000:GOTO 210
240 R=1:FOR I=N TO (N-K+1) STEP -1
250 R=R*I:NEXT I
260 R=R/FACT K
270 PRINT"NUMERO COMBIN. =":PRINT R:END
300 "D":PRINT " * COMB. CON RIP. * "
310 INPUT "NUM. OGGETTI ?";N:IF N<1 THEN
  GOSUB 1000:GOTO 310
320 INPUT "CLASSE ?";K:IF K<1 THEN GOSUB
  1000:GOTO 320
330 R=1:FOR I=(N+K-1) TO N STEP -1
340 R=R*I:NEXT I
350 R=R/FACT K:GOTO 270
400 "F":PRINT " * PERMUTAZIONI * "
410 INPUT "NUM. OGGETTI ?";N:IF N<1 THEN
  GOSUB 1000:GOTO 410
420 R=FACT N
430 PRINT"NUMERO PERMUT. =":PRINT R:END
1000 PRINT "ERRORE IN INPUT":RETURN
```




È ormai un mese che è cominciata la stagione calcistica 1987/1988 e, con essa, il giro di miliardi che accompagneranno i concorsi del totocalcio fino al prossimo giugno.

Il fenomeno ha ormai, da alcuni anni a questa parte, contagiato numerosissimi possessori di home e personal computers, che, consapevoli delle ampie possibilità che offrono queste macchine più o meno potenti, hanno creato un vero e proprio software dedicato sistemi di riduzione per il totocalcio.

Il programma che vi presentiamo, molto semplice e senz'altro suscettibile di miglioramenti, vi dà la possibilità di giocare una schedina con sei doppie ed ottenere una riduzione da 64 a 20 colonne effettive da giocare.

Dopo aver introdotto la data della schedina da giocare, il computer vi richiederà i nomi delle 26 squadre degli incontri previsti dalla schedina stessa, che dovrete immettere seguendo le istruzioni del vostro calcolatore.

Completata questa operazione, sarà visualizzata la schedina in questione con la possibilità di immettere i pronostici, uno per uno. Anche questa fase del programma è completamente guidata dal computer che vi ricorderà, man mano che voi procederete con i pronostici, quante doppie vi restano ancora da giocare.

Terminata, con l'immissione dell'ultimo pronostico, la fase di INPUT, ecco che il computer visualizzerà la schedina con tutte e 20 le colonne ottenute, con una riduzione standard, dalle 64 originali.

Coloro che sono in possesso di una stampante, infine potranno ottenere su carta la schedina, esattamente come compare sul monitor.

STRUTTURA DEL PROGRAMMA

- 10-40** : DIMENSIONAMENTO VARIABILI. Vengono dimensionate le variabili del programma.
- 50-220** : PRESENTAZIONE. Una schermata vi spiega il programma, che procede con l'immissione della data della schedina.
- 230-380** : 1^a SCHERMATA-INPUT SQUADRE. Il computer vi chiede di immettere, una per una, le 26 squadre della schedina.
- 390-620** : 2^a SCHERMATA-INPUT PRONOSTICI. Ora la schedina in questione è rappresentata sul video. Il computer vi guida nell'immissione dei pronostici.
È richiamata la routine delle righe 10000-10040.
- 630-830** : 3^a SCHERMATA-OUTPUT VIDEO. Il computer elabora la riduzione delle 64 colonne, e visualizza le 20 così ottenute. Potete scegliere di stampare le colonne stesse.
- 840-980** : OUTPUT STAMPANTE. Viene stampata la schedina da giocare con le 20 colonne ottenute dalla riduzione del computer.
- 1000-7040** : ROUTINES ELABORAZIONE DATI. Vengono elaborate le 6 doppie e le 7 fisse da giocare sulla schedina.
- 10000-10040** : VISUALIZZAZIONE PRONOSTICI. Durante la fase di immissione dei pronostici, questa routine provvede alla visualizzazione dei pronostici stessi.

1

ST 10/87

2

▶▶▶

3

4



LIST 10/87

Tecniche di computergrafica



di O. Contenti

Il superprogramma grafico

Di certo ricorderete come già nella scorsa puntata abbiamo teso a rilevare la pressoché completa implementabilità dei listati pubblicati in questa rubrica di grafica; ebbene, si dà il caso che il nostro coordinatore tecnico, una specie di post-vichingo dalla chioma dorata, l'abbia voluta mettere alla prova. Il Nicolai Fabrizio (questo il nome che gli dette Odino), forte della memorizzazione di circa 1500 linguaggi di programmazione, pose i suoi 10 terminali (le dita) sulla tastiera dell'Atari 520, cominciò a digitare alla velocità della luce, ma colto da improvviso raptus stese un listato con istruzioni Fortran miste a Pascal con routinette in linguaggio macchina! Dette un RUN con sei U fra lo sbigottimento dei colleghi di redazione, quindi l'Atari esplose in mille pezzi, rendendo un Nicolai riccioluto dallo aspetto vagamente nigeriano! Lo portammo alla informatic-deliri mentre ripeteva ancora: «Voglio il PEPPY mv 916 senza abbonamento!», mentre la segretaria Serena gli accarezzava la fronte sudata. Voglio dire che non va a finire sempre in questo modo con i nostri programmi, comunque attenzione: non si sa mai!

Un invito ai lettori!

Avete qualche routine grafica che ritenete interessante? Se questa non è eccessivamente lunga (non oltre le 30 righe) e assolutamente INEDITA, vale a dire mai apparsa su riviste, libri o manuali del settore informatico, inviatela a:

LIST. Rubrica «AIDED STYLING» Via Flavio Stilicone 111, 00175 Roma;

completa di LISTATO o relativa spiegazione, in breve, del suo contenuto. In alternativa al listato potre-

te sempre inviarci una cassetta con miniprogramma; le creazioni grafiche più degne di nota verranno pubblicate assieme al nome dell'autore nelle pagine di questa stessa rubrica. I listati possono anche essere delle semplici, ma interessanti variazioni dei programmi editi in queste stesse pagine, in questo caso ricordate di segnalarci la puntata della quale avete attinto i vostri spunti, non inviateci le sole varianti, ma il listato completo.

TAVOLA DELLE ISTRUZIONI EQUIVALENTI

Funzioni	Sistemi MSX	SIMON C64	CBM 128	OLIVETTI PC 128	SEGA SC 3000	ATARI 520	SPECTRUM ZX 48 K
Punti	PSET	PLOT	DRAW	PSET	PSET	LINEF	PLOT
Linee	LINE	LINE	DRAW	LINE	LINE	LINEF	DRAW
Riempie	PAINT	FILL	PAINT	PAINT	PAINT	PAINT	****
Cerchi	CIRCLE	CIRCLE	CIRCLE	CIRCLE	CIRCLE	CIRCLE	CIRCLE
Blocchi pieni	LINEBF	BLOCK	BOX	BOXF	(PAINT)	(FILL)	****
Blocchi vuoti	LINEB	(LINE)	BOX	BOX	(LINE)	(LINEF)	****

Legenda: (PAINT) = le istruzioni fra parentesi possono simulare con qualche variazione la funzione desiderata

— **** = istruzione non presente.

SPIRALI, GIRANDOLE E S

"CONTINUA UNO DEI PIÙ AFFASCINANTI VIAGGI NEL MONDO DE



con una tavoletta grafica e... un po' di fantasia

Spirali, girandole e serpentine

Vi accorgerete di quali figurazioni fantasiose ed avvincenti sia capace il vostro HOME, descrizioni grafiche che avete spesso ammirato sui monitor degli shop di informatica e credevate difficilmente realizzabili col vostro elaboratore. La rubrica grafica di LIST ora vi rende tutto più agevole, anche per il fatto di non sottoporvi una lunga serie di listati differenti, ma per quella che ormai è divenuta una saggia regola, un solo listatino di base sul quale ruotano minute e semplici VARIANTI che si allacciano al tema grafico del mese. Tema che in questa seconda puntata, come da titolo, illustra il modo per realizzare diversi tipi di girandole, circolari o squadrate, spirali concentriche, serpentine, ma anche stelle e vortici.

Come al solito il list di base e le varianti verranno date in BASIC-MSX, ma sia la tavola delle istruzioni equivalenti che le spiegazioni sulla sintassi delle righe di istruzioni adottate, rendono questa rubrica onnicomprensiva per una vasta gamma di HOME, anche di quelli non citati nella tabella di cui sopra.

Sintassi delle istruzioni grafiche MSX.

La sintassi serve per implementare correttamente le seguenti istruzioni con quelle del vostro HOME.

— **LINE** (colonna di partenza, riga di partenza) — (colonna di arrivo, riga di arrivo), colore, box

Se «box» è «BF» (come nella variante C) vuol dire che viene disegnato un quadrilatero pieno che ha per diagonale le coordinate di partenza con quelle di arrivo.

Quindi se non avete l'opzione BF fate disegnare il quadrilatero riempiendolo con una successiva istruzione equivalente a **PAINT** (vedi tabella).

Ancora, se non esiste l'opzione «box» (tutte le altre varianti), verrà descritta una linea che parte dalle coordinate di partenza fino a quelle di arrivo.

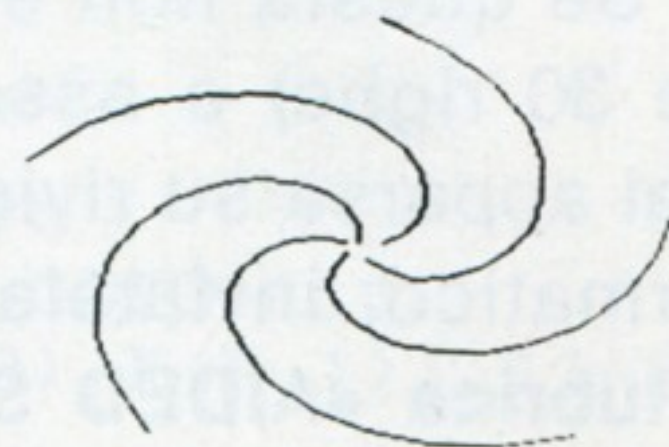
— **CIRCLE** (colonna centro, riga centro), raggio

— **PAINT** (colonna, riga)

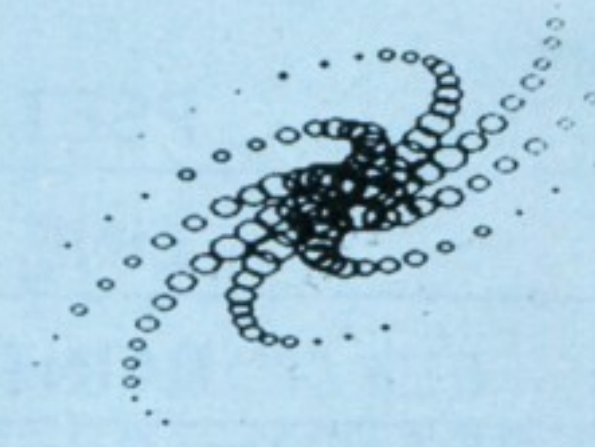
Nella **PAINT** il riempimento della figura inizia dalla colonna e dalla riga indicata



LISTATO STANDARD



C1



G

SERPENTINE

LA GRAFICA COMPUTERIZZATA"

MSX AIDED
STYLING



cg

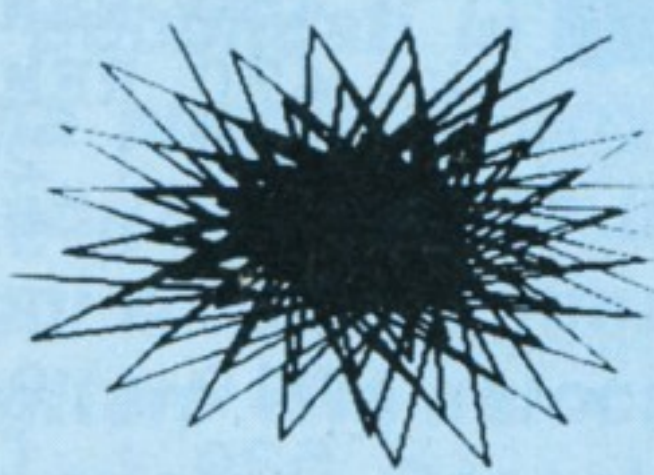
2

di O. Contenti

Tecniche di computergrafica



A



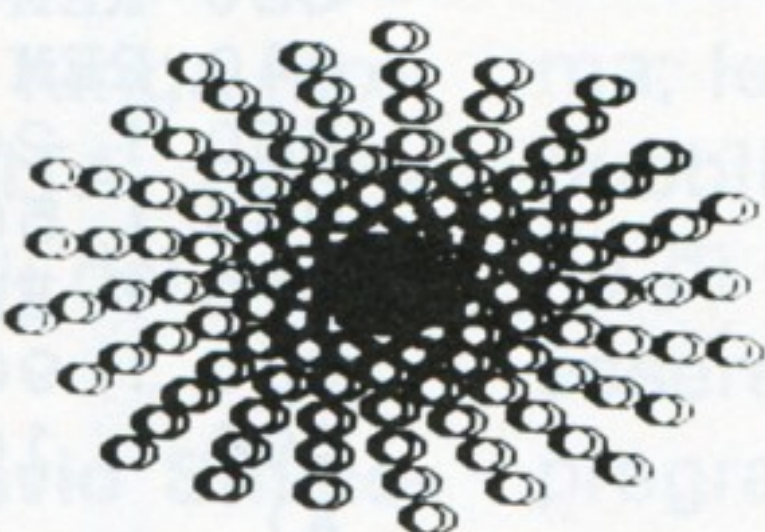
B



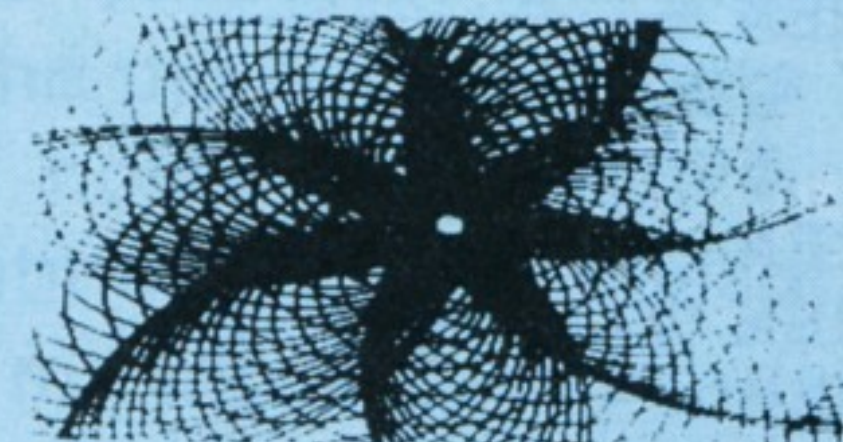
C



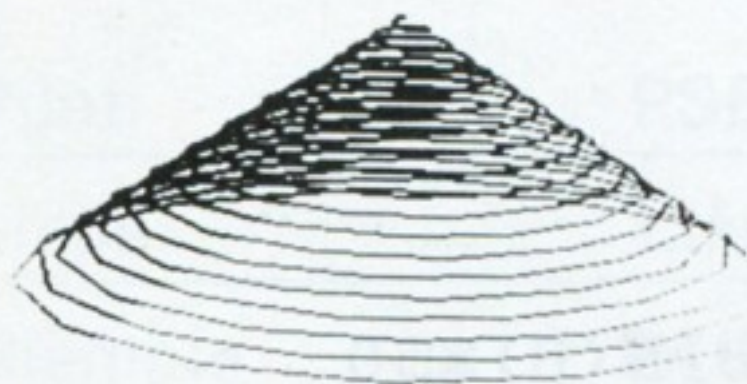
D



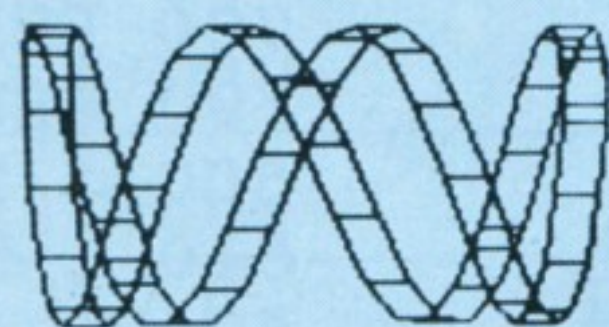
E



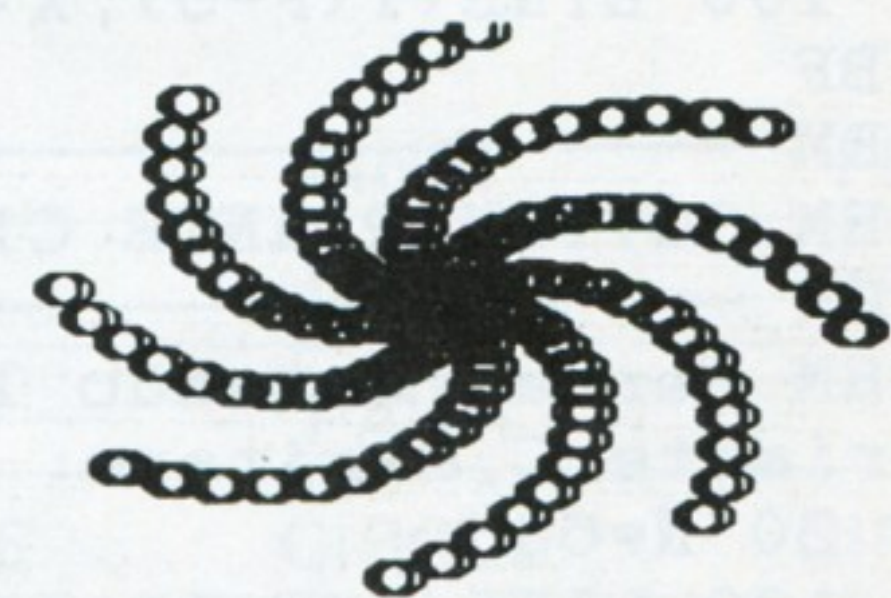
F



H



I



VARIANTE EXTRA LISTATO

LISTATO BASE e VARIANTI



di O. Contenti

Tecniche di computergrafica

Avvertenze

Prima di passare alle varianti ricordiamo che: SCREEN 2 sta per schermo in alta risoluzione (191-255); COLOR 1,11,11 pone la figura in nero su sfondo e cornice gialle; CLS pulisce lo schermo.

Le varianti

Dopo aver adottato ogni variante, ripristinate il listato base originale per ottenere gli effetti grafici desiderati.

2

```

1 REM
2 REM Computergrafica 2
3 REM Spirali Girandole
4 REM e Serpentine
5 REM
6 REM LISTATO BASE
7 REM
8 REM di O. Contenti
9 REM

10 SCREEN 2:COLOR 1,11,11:CLS
20 N=440
30 DIM Y(N),X(N)
40 FOR F=1 TO N
50 A=A+.2
60 C=COS(A)*A+120:R=SIN(A)*A+100
70 PSET (C,R)
80 Y(F)=C:X(F)=R
90 IF F=1 THEN GOTO 110
100 LINE (Y(F-1),X(F-1))- (Y(F),X(F))
110 NEXT F
120 GOTO 120
130 REM
140 REM VARIANTI
150 REM
160 REM
165 REM
170 REM VARIANTE A
180 REM
190 ! 20 N=78
200 ! 50 A=A+1.5
205 REM
210 REM VARIANTE B
220 REM
230 ! 20 N=98
240 ! 50 A=A+1.2
250 ! ELIMINARE RIGA 70
260 ! 90 IF F<5 THEN GOTO 110
270 ! 100 LINE(Y(F-4),X(F-4))- (Y(F-1),X
(F-1))
275 REM
280 REM VARIANTE C
290 REM
300 ! 20 N=150
310 ! 50 A=A+1.3
320 ! 90 IF F<6 THEN GOTO 110
330 ! 100 LINE(Y(F-5),X(F-5)), (Y(F),X(F
)),1,BF
335 REM
340 REM SOTTOVARIANTE C1
350 REM
360 REM Ferme restando le modifiche del
la variante C, digitare:
370 ! 20 N=68
380 ! 100 LINE(Y(F-5),X(F-5))- (Y(F),X(F
))
385 REM
390 REM VARIANTE D
400 REM
410 ! 20 N=91

```

```

420 ! 50 A=A+1.2
430 ! 90 IF F<12 THEN GOTO 110
440 ! 100 LINE (Y(F-10),X(F-10))- (Y(F),
X(F))
450 ! 101 LINE (Y(F-10),X(F-10))- (Y(F-1
),X(F-1))
455 REM
460 REM VARIANTE E
470 REM
480 ! 20 N=151
490 ! 50 A=A+.6
500 ! ELIMINARE LA RIGA 70
510 ! 90 IF F<4 THEN GOTO 110
520 ! 100 CIRCLE(Y(F-3)+5,X(F-3)),5
530 ! 101 CIRCLE(Y(F-3)+8,X(F-3)),5
540 ! 102 PAINT (Y(F-3)+12,X(F-3))
545 REM
550 REM VARIANTE F
560 REM
570 ! 20 N=151
580 ! 50 A=A+.9
590 ! 60 C=COS(A)*A+125:R=SIN(A)*A+100
600 ! ELIMINARE RIGA 70
610 ! 90 IF F<4 THEN GOTO 110
620 ! 100 CIRCLE(Y(F-1),X(F-1)),5+A
625 REM
630 REM VARIANTE G
640 REM
650 ! 20 N=150
660 ! 50 A=A+.8
670 ! ELIMINARE RIGA 70
680 ! 90 IF F<10 THEN GOTO 110
690 ! 100 CIRCLE(Y(F-2),X(F-1)),11-SQR(
A)
695 REM
700 REM VARIANTE H
710 REM
720 ! 20 N=227
730 ! 50 A=A+.5:B=B+.173
740 ! 60 C=COS(A)*A+130:R=SIN(A)*B+170-
A
750 ! ELIMINARE LA RIGA 70
760 ! 90 IF F<5 THEN GOTO 110
770 ! 100 LINE (Y(F),X(F))- (Y(F-1),X(F-
1))
775 REM
780 REM VARIANTE I
790 REM
800 ! 20 N=52
810 ! 50 A=A+.12:B=B+.5
820 ! 60 C=COS(A)*60:R=SIN(B)*50
830 ! 70 LINE (C+120,R+80)- (C+130,R+80)
840 ! 80 Y(F)=C+120:X(F)=R+80
850 ! 90 IF F=1 THEN GOTO 110
860 ! 100 LINE (Y(F-1),X(F-1))- (Y(F),X(
F))
870 ! 101 LINE (Y(F-1)+10,X(F-1))- (Y(F)
+10,X(F))

```


AUTOPROGRAMMA GRAFICO ■■■

Un programma che... ne crea un altro decodificando in tempo reale i disegni eseguiti su schermo, tramite stick o tastiera.

Cos'è l'Autoprogramma Grafico

È un programma che — senza presunzione — rivoluziona il concetto dei software dedicati agli Home Computers.

Infatti, quando utilizziamo un normale programma sappiamo sin dall'inizio che questo non modificherà mai nulla di sé stesso e per quanto possa essere complesso e variegato mai ci aspetteremmo che lo stesso aggiunga, auto-creandole, righe di listato non presenti alla ditta.

Invece «Autoprogramma Grafico» cresce di numeri di linea man mano che lo usate, traducendo in righe Basic tutto quello che state disegnando sul video.

Ovviamente, quando avrete deciso di dare fine al disegno, tutte quelle linee di programma saranno perfettamente registrabili come se fossero sempre appartenute al programma madre.

Quindi l'Intelligenza Artificiale entra di prepotenza nella casa degli HOME (scusate il bisticcio), e non a caso a proporvelo è LIST, così mentre le riviste concorrenti vi sommergono solo di prove Hardware o dell'ennesimo «filetto» al calcolatore, noi vi offriamo delle brucianti novità pronte per l'uso.

L'evoluzione di un'idea

Già da qualche mese, assieme all'amico Fabrizio Nicolai (il nostro Coordinatore Tecnico), rubando tempo al sonno e alle notti romane, avevamo concordato degli appuntamenti serali.

In compagnia di PC professionali abbiamo ricercato e studiato diversi interventi di I.A. in programmi di varia utilità; fra i risultati un «Deduttore di Algoritmi» ancora in cantiere ma di ottima stesura.

Fu in quelle notti che mi resi conto, attraverso l'analisi degli esperimenti compiuti, della mancanza pressoché assoluta di software intelligenti ad indirizzo grafico per Home.

L'Autoprogramma Grafico era stato sempre un mio sogno (anche se non lo chiamavo ancora così), mi sono lanciato

a capofitto nella nuova impresa e non certo con la sicurezza di riuscire nell'intento.

Ma gli indugi teorici hanno lasciato la strada alla pratica e il risultato è ora al vostro giudizio.

L'idea portante di tutto il programma è stata di far creare dallo stesso elaboratore delle stringhe (nel list sempre X(Z)) riproducenti e simulanti perfettamente: numeri di linea con relative istruzioni grafiche che traducevano istante per istante le immagini video impostate dall'utente, a questo punto non rimaneva che stampare e far accettare come veri numeri di linea queste stringhe-istruzioni, cosa che avviene puntualmente nel corso del programma.

Detto così tutto appare elementare, ma rispettare la sintassi di ogni istruzione, la lunghezza massima delle linee, seguire passo passo per memorizzare con la DRAW tutti i movimenti possibili e passare ad un nuovo numero di linea quando occorre (per citare solo parte dell'algoritmo), non è stato impegno da poco.

A chi è indirizzato e a che cosa serve l'APG

L'impiego dell'Autoprogramma Grafico è multidirezionale nel senso che può essere sfruttato sia da un accanito cultore di videogames (per approntare favolosi e complessi sfondi di gioco senza dover precedentemente mappare su carta il disegno per poi faticosamente tradurlo in Basic) che da un professionista che intenda memorizzare in un vero programma i suoi progetti o i suoi estemporanei architettonici o geometrici, il tutto anche senza conoscere una sola riga di Basic!

Il risparmio di tempo poi risulta essenziale e quello che una volta avreste realizzato in ore e alle volte in giorni, ora con l'APG vi verrà reso giusto nel tempo impiegato per produrre il disegno in video, un programma che attraverso dei semplici MERGE potrà essere collegato a qualsiasi altro software.

Lavorando di meno e sprecando minori energie le vostre realizzazioni grafiche acquisteranno quella importanza e complessità che avrete sempre desiderato, ma che non ha



mai avuto modo di esprimersi causa la lungaggine della traduzione in Basic delle immagini video, un vero balzo di qualità per la vostra produzione di immagini che ora può prendere il volo.

Utilizzazioni future dell'APG

In redazione già si parla di pacchetti software per videogames, dove tutto, dalla costruzione degli sfondi alla creazione di sprite fino alla loro animazione viene tradotto in programma da relativi APG, avendo in mano questi programmi chiunque avrà a disposizione un «fai da te» di listati pronti e regalarvi le emozioni di un videogame, esclusa la fatica per produrlo.

Ancora, studieremo degli APG applicati al linguaggio e qui la simbiosi con il software didattico avverrà quasi naturalmente.

Ma è ancora troppo presto per dirlo, l'APG è appena nato e quindi ha bisogno di ulteriori studi e ricerche di approfondimento per avanzare, nonché dei vostri spunti e suggerimenti che saremo felicissimi di accogliere.

Tastiera Stick

È in verità relativamente complessa, ma se pensiamo che indirettamente fa produrre un listato che noi stessi realizzeremo solo in diverse ore, ci rendiamo conto che il gioco vale la candela.

— Le 8 direzioni del tratto

Possono essere comandate sia da STICK che da TASTIERA, anche nel caso di distacco della «penna», quindi:

■ **Lo STICK** (della porta uno) muoverà il tratto nelle 8 direzioni usuali.

■ **I 4 tasti CURSORE** imiteranno lo stick nei movimenti: in ALTO; in BASSO; a DESTRA e a SINISTRA.

— Le direzioni diagonali faranno capo ai TASTI FUNZIONE:

■ **F1** per dir. in ALTO a SINISTRA.

■ **F2** per dir. in ALTO a DESTRA.

■ **F3** per dir. in BASSO a DESTRA.

■ **F4** per dir. in BASSO a SINISTRA.

Ogni movimento sarà costantemente seguito da 2 MIRINI-SPRITES sovrapposti (di diverso colore) per non perdere mai di vista l'ultima posizione raggiunta dal tratto che si pone sempre ancor di più nel caso di distacco della penna.

— Per il DISTACCO della PENNA

■ Premere il tasto **ESC** (della tastiera) o il pulsante di FUOCO (● FIRE) dello stick, in entrambi i casi ora vi esporrete senza disegnare nulla e raggiunta la nuova posizione desiderata premere nuovamente gli stessi ESC o FIRE per descrivere un nuovo tratto.

— Tasti abilitati con PENNA ATTACCATA

■ **Tasto 1:** Cambia a vostro piacere il colore del tratto fra i 15 consentiti, il colore viene visualizzato da un apposito SPRITE posto in alto a destra dello schermo che ne scandisce le variazioni seguendo la numerazione usuale dei codici colore.

■ **Tasto 2:** Accetta e memorizza nell'Autoprogramma il colore scelto con il tasto 1, se il tasto 2 non viene premuto il tratto cambierà ugualmente colore ma questa variazione non farà parte dell'autoprogramma. Ricordate quindi di premere il tasto 2 dopo aver scelto un nuovo colore.

■ **Tasto 3:** Crea la funzione PAINT (riempimento figure) a cominciare dal centro dei MIRINI-SPRITES. Per farlo funzionare correttamente: premere ESC o FIRE prendere nuova posizione, ripremete ESC o FIRE e solo allora premere il tasto 3.

— Tasti abilitati con PENNA STACCATA

■ **Tasto 4:** Traccia una linea dall'ultimo punto lasciato alla nuova posizione raggiunta, tale funzione immessa in APG (autoprogramma), in modo M x,y come da DRAW. Per attivarla premere ESC o FIRE raggiungere il punto desiderato, quindi premere il tasto 4, ora la penna è riattaccata allo schermo.

■ **Tasto 5:** Traccia il perimetro di un BOX. Procedere come per il tasto 4.

■ **Tasto 6:** Traccia un BOX pieno. Procedere come sopra.

■ **Tasto 7:** Traccia una circonferenza. Per realizzarla premere ESC o FIRE spostarsi di colonna (o a destra o a sinistra) e premere il tasto 7, la distanza percorsa sarà relativa al raggio del cerchio. Alla stampa della circonferenza la penna sarà nuovamente attaccata.

■ **Tasto "(" = shift + 9:** Dà lo STOP al disegno e passa forzatamente alla STAMPA dell'AUTOPROGRAMMA, condizione che si verifica anche in un altro caso come approfondiremo nel seguente capitoletto.

La stampa dell'Autoprogramma

Quello che potrete comandare con il tasto "(" (ricordiamo che risulta attivo solo con penna attaccata), verrà invariabilmente provocato dal programma stesso ogni 5-6 numeri di linea indirettamente prodotti dal vostro disegno.

È in questa parte del programma (righe 1640-1770) che il listato evidenzia tutto il lavoro interno, infatti senza preavviso, mentre disegnate, la pagina grafica passerà al MODO TESTO zero, visualizzando delle righe di programma a partire dalla riga 2000.

Ma non fatevi ingannare, in realtà esse non sono ancora righe di programma, ma STRINGHE che simulano in maniera precisa le stesse.

A questo punto l'MSX ha bisogno di voi per mutarle in vere righe di listato, capito il trucco?

Quindi portare il cursore sul PRIMO NUMERO di LINEA e premere RETURN per tutte le linee presentate sullo schermo, fatto questo se date il LIST 2000 vedrete effettivamente le righe far parte «magicamente» del listato.

Ora, non vi resta che dare il RUN 2000 per ritornare fulmineamente alla pagina grafica e ritrovarvi il disegno che avevate lasciato.

Questa routine: RETURN per ogni linea e RUN 2000 dovrà essere eseguita ogni volta per incamerare la nuova parte del programma.

Naturalmente i numeri di linea del programma verranno incrementati ad ogni passaggio, righe che potrete editare come quelle normali o non accettare (non battendo RETURN) nel caso ad esempio di un PAINT disastroso, quante volte capita!

Premendo il tasto per "(" avrete le medesime condizioni, ma magari, proprio perché bloccata da voi, vi ritroverete anche una sola linea da memorizzare, ma questo è a vostra scelta.

Nella stampa dell'APG, troverete sempre una strana riga, la 165, essa è di importanza fondamentale per la riuscita di tutto il programma, questa infatti memorizza tutte le variabili che andrebbero perse in seguito all'immissione in memoria di qualsiasi nuova riga di programma, invece, in questo modo la 165 rammenta: il colore schermo (S\$); il colore del tratto (P\$); il prossimo numero di linea per l'APG (NL); le ultime coordinate del MIRINO (C ed R) e l'ultimo colore usato (T).

Avvio e conclusione del programma

Dopo il titolo, scegliere come richiestovi il colore dello SCHERMO e il colore del TRATTO, ma ricordo che ambedue potranno essere mutati agendo nel corso del programma: il primo con un PAINT per l'intero schermo e il secondo con i tasti 1 e 2.

Impostati i colori, il programma partirà visualizzando il MIRINO nelle coordinate 100,100 e lo SPRITE segnala colore.

Quando deciderete di dare termine al vostro disegno premete il già citato tasto "(", battere RETURN per ogni linea e mutare la riga 10000 in:
10000 GOTO 10000.

Ora se volete registrare PROGRAMMA ed AUTOPROGRAMMA fatelo pure, ma se volete mantenere solo il disegno battete il comando: DELETE 10-1810 ottenendo il solo AUTOPROGRAMMA pulito, quindi registrate normalmente.

Non rinumerate il listato

Lo rovinereste irreparabilmente e anche se vi pare strana la indicazione della linea 10000 che riporta ad un GOTO 165, lasciatela com'è, potrete rinumerare il tutto (senza pregiudicare nulla) solo dopo aver completato il disegno in fase di registrazione su nastro o disco.

Avvertenze e consigli

— Ogni volta che premete un tasto otterrete il «ricevuto» da un segnale acustico, ma non indugiate troppo sui tasti

(che non siano di movimento) perché potreste causare delle inutili ripetizioni nell'APG.

— I tasti funzione: F1, F2, F3 ed F4 sono stati opportunamente ridefiniti, quindi i comandi relativi vanno solo digitati.

— Mentre disegnate avvertirete ogni tanto un BEEP, esso vi segnala che l'elaboratore ha completato una riga dell'APG e ne sta cominciando un'altra.

— Se dopo aver memorizzato alcune righe di APG avete digitato un RUN senza numero di linea, interrompete il programma e battete RUN 2000, non avrete causato nessun danno al programma.

— Per cancellare tratti non desiderati impostate il colore di sfondo ripassando il segno o fate uso del PAINT.

— Il programma ha fine solo con un MEMORY FULL (se non lo concludete prima), ma niente paura, per l'APG c'è posto per un listato di lunghezza colossale, ho fatto uso di soli 7000 Bytes.

— Tutte le REM possono essere omesse nella digitazione.

Conclusione

Qualcuno potrebbe obiettare: Perché creare un AUTOPROGRAMMA quando per memorizzare una immagine possiamo usare un BLOAD in VRAM o un caricamento in FILE?

La risposta è semplice, primo, perché un'immagine compressa in FILE o tratta dalla VRAM è molto difficile da mutare per una qualsiasi variazione anche minima del disegno, mentre con un listato tutto risulta davvero elementare, secondo, perché la comprensione di questa grafica intelligente ci apre le porte a panorami ben più vasti di una codifica in file, il discorso in effetti si apre ora sulle possibili aree di sfruttamento di una simile idea.

L'intelligenza artificiale non è solo una moda, ma una realtà che ci investe a ondate, un'appassionante ricerca che vivremo in un futuro molto prossimo, una realtà che possiamo cominciare a comprendere anche attraverso il nostro piccolo, ma prezioso MSX!

VARIABILI PRINCIPALI

Gi	se G = 1 = inizio disegno; se G = 2 = disegno già iniziato
Z	indice array stringhe contenenti le righe simulate
XS(Z)	= contiene le linee APG simulate
C,R	= coordinate mirino
NL	= numero di linea per APG
K\$ o KK\$	= richiami da tastiera
J o JJ	= direzioni stick
W e V	= precedente ed ultima direzione di tratto scelta
L	= lunghezza di una funzione DRAW
F \$	funzione DRAW
T	= colore tratto
RC	= raggio cerchio
CO,RI	= coordinate di arrivo per circle; line b o bf; M x,y
C1 e C2	= colore primo e secondo sprites-mirini
P o PP	= richiamato tasto ESC o pulsante FIRE

STRUTTURA DEL PROGRAMMA

40/130	titolo e input per colore schermo e tratto
160/320	iniz. screen 2 + sprites cursori e sprite segnala-colore
165	linea non presente nella digita ma che sarà creata dal programma stesso
330/360	stringhe primi 2 numeri di linea APG
400/440	input per stick e tastiera con penna attaccata
470/620	le 8 direzioni di tratto con penna attaccata
650	stampa del tratto eseguito
680/750	routine dopo richiamo tasto ESC o FIRE
780/840	input per stick e tastiera con distacco penna
850/920	le 8 direzioni di tratto con distacco penna
930	mirini-sprites seguenti gli spostamenti
950/1020	routine riattacco penna
1050	tracciamento linee
1080/1130	mem. funzione LINE B o BF in APG
1160/1190	mem. funzione CIRCLE in APG
1220/1250	mem. funzioni DRAW in APG
1290/1300	mem. funzioni DRAW + incremento numero linea per APG
1330/1340	solo cambio colore tratto
1370/1440	mem. colore scelto in APG
1470/1510	mem. con incremento numero di linea per colore scelto in APG
1540/1610	mem. funzione PAINT con incremento numero di linea in APG
1640/1770	preparazione e stampa APG
da 2000	in poi APG non presente alla digita ma che verrà creato in seguito dal programma stesso
10000	linea che si indirizza ad una linea per ora inesistente (la 165) ma che verrà creata in seguito dal programma stesso

Un esempio di programma scritto interamente dall'APG

Per dare prova delle qualità del nostro APG vi proponiamo un listato su cui il programmatore non ha modificato né aggiunto o alterato qualsiasi riga di programma. Il solo intervento che è stato effettuato è stato quello di un RETURN generale che fa partire il programma dalla riga 10 anziché dalla riga 2000. L'immagine che apparirà sullo schermo a seguito della digita, sarà quella di una veduta stradale con uno sfondo di caseggiati finemente disegnati.

In conclusione, per i cultori dell'immagine, proponiamo l'inserimento di 2 numeri di linea che agendo sul VDP riescono a far stampare l'intera immagine, non più pezzo per pezzo come da disegno eseguito, ma in una fulminea visualizzazione di tutta l'area grafica occupata.

I numeri di linea sono i seguenti:

15 VDP(1) = VDP(1)AND&HBF

695 VDP(1) = VDP(1)OR&H40

LIST

1

```

10 'AUTOPROGRAMMA GRAFICO
20 '
30 '—— di O. Contenti ——
40 SCREEN 1:KEYOFF:CLS
50 LOCATE 0,10:PRINTTAB(4);"Autoprogramma Grafico":PRINT:PRINTTAB(5);"di:Oswaldo Contenti":PRINT:PRINTTAB(10);"Ediz.LIST"
60 FORTM=1TO5000:NEXTTM:CLS
70 CLEAR 1000
80 PRINT:PRINT"Colore schermo (1/15)"
90 INPUT S$
100 IF VAL(S$)<1ORVAL(S$)>15THEN CLS:GOTO 80
110 PRINT:PRINT"Colore tratto (1/15)"
120 INPUT P$
130 IF VAL(P$)<1ORVAL(P$)>15THEN CLS:GOTO 110
140 REM SCHERMO GRAFICO
150 REM
160 SCREEN 2,0,0:COLOR VAL(P$),VAL(S$),VAL(S$):CLS
170 G=1:Z=1:H=0
180 PLAY "V6L6404S0M5000"
190 IF Y=1 THENGOTO 220
200 T=VAL(P$):C=100:R=100
210 REM SPRITE-COLORE
220 FOR TM=14336 TO 14343:VPOKE TM,255:NEXT TM:PUTSPRITE 0,(247,0),T
230 REM SPRITES-CURSORI
240 IF VAL(S$)>1 AND VAL(S$)<14 THEN C1=1:C2=15:GOTO 260
250 IF VAL(S$)=1 THEN C1=2:C2=15 ELSE C1=1:C2=4
260 DATA 0,0,0,238,0,0,0,0,16,16,16,0,16,16,16,0
270 FOR TM=14344 TO 14359
280 READ A
290 VPOKE TM,A:NEXT TM
300 PUTSPRITE1,(C-3,R-4),C1:PUTSPRITE2,(C-3,R-4),C2
310 DIM X$(9)
320 IF Y=1 THENGOTO 350
330 X$(0)="2000 SCREEN 2,0,0:COLOR "+P$+"", "+S$+", "+S$+":CLS"
340 NL=2010
350 C$=STR$(C):R$=STR$(R)
360 X$(Z)=STR$(NL)+CHR$(32)+"DRAW "+CHR$(34)+"BM"+MID$(C$,2,LEN(C$))+", "+MID$(R$,2,LEN(R$))
370 KEY1,"$":KEY2,"%":KEY3,"&":KEY4,"'"
380 REM TASTIERA E STICK
390 REM
400 P=STRIG(1):K$=INKEY$:J=STICK(1)
410 ON P+2 GOTO 680,420
420 IF K$="" THENGOTO440
430 IF ASC(K$)<27 OR ASC(K$)>51 THEN GOTO 440 ELSE ON ASC(K$)-26 GOTO 680,510,590,470,550,440,440,440,440,610,490,530,570,1640,400,400,400,400,400,400,400,400,1330,1370,1540
440 IF J=0 THEN GOTO 400 ELSE ON J GOTO 470,490,510,530,550,570,590,610
450 REM LE 8 DIREZIONI
460 REM
470 IFR=0 THENGOTO400ELSEW=V:V=1:IFW<>VT HENGOTO1220
480 L=L+1:F$="U":R=R-1:GOTO 650
490 IFR=0ORC=246 THENGOTO400ELSEW=V:V=2:IFW<>V THENGOTO1220

```


LIST

2

```

500 L=L+1:F$="E":R=R-1:C=C+1:GOTO 650
510 IFC=246THENGOTO400ELSEW=V:V=3:IFW<>
VTHENGOTO1220
520 L=L+1:F$="R":C=C+1:GOTO 650
530 IFC=246ORR=191THENGOTO400ELSEW=V:V=
4:IFW<>VTHENGOTO1220
540 L=L+1:F$="F":C=C+1:R=R+1:GOTO 650
550 IFR=191THENGOTO400ELSEW=V:V=5:IFW<>
VTHENGOTO1220
560 L=L+1:F$="D":R=R+1:GOTO 650
570 IFC=0ORR=191THENGOTO400ELSEW=V:V=6:
IFW<>VTHENGOTO1220
580 L=L+1:F$="G":C=C-1:R=R+1:GOTO 650
590 IFC=0THENGOTO400ELSEW=V:V=7:IFW<>VT
HENGOTO1220
600 L=L+1:F$="L":C=C-1:GOTO 650
610 IFC=0ORR=0THENGOTO400ELSEW=V:V=8:IF
W<>VTHENGOTO1220
620 L=L+1:F$="H":C=C-1:R=R-1:GOTO 650
630 REM STAMPA TRATTO
640 REM
650 PSET(C,R),T:PUTSPRITE1,(C-3,R-4),C1
:PUTSPRITE2,(C-3,R-4),C2:GOTO400
660 REM Tasto FIRE o ESC
670 REM
680 PLAY"EFC":IF LEN(X$(Z))>=150 OR H=1
THEN GOTO 710
690 IF F$=""ORL=0 THEN GOTO 750
700 L$=STR$(L):X$(Z)=X$(Z)+F$+MID$(L$,2
,LEN(L$)):H=0:GOTO 750
710 IF H=1ANDL=0THENGOTO750
720 IF H=1ANDL>0THENGOTO 740
730 X$(Z)=X$(Z)+CHR$(34)
740 Z=Z+1:NL=NL+1:L$=STR$(L):X$(Z)=STR$
(NL)+" DRAW "+CHR$(34)+F$+MID$(L$,2,LEN
(L$)):H=0
750 FORTM=1TO500:NEXTTM:GOTO780
760 REM DISTACCO PENNA
770 REM
780 CO=C:RI=R
790 PP=STRIG(1):KK$=INKEY$:JJ=STICK(1)
800 ON PP+2 GOTO 960,810
810 IF KK$=""THENGOTO840
820 IF ASC(KK$)>51 AND ASC(KK$)<56 THEN
ON ASC(KK$)-51 GOTO 970,1080,1090,1160
830 IF ASC(KK$)<27 OR ASC(KK$)>39 THEN
GOTO 840 ELSE ON ASC(KK$)-26 GOTO 960,8
70,910,850,890,840,840,840,840,920,860,
880,900
840 IF JJ=0 THEN GOTO 790 ELSE ON JJ GO
TO 850,860,870,880,890,900,910,920
850 IFR=0THENGOTO790ELSER=R-1:GOTO930
860 IFR=0ORC=246THENGOTO790ELSER=R-1:C=
C+1:GOTO930
870 IFC=246THENGOTO790ELSEC=C+1:GOTO930
880 IFC=246ORR=191THENGOTO790ELSEC=C+1:
R=R+1:GOTO930
890 IFR=191THENGOTO790ELSER=R+1:GOTO930
900 IFC=0ORR=191THENGOTO790ELSEC=C-1:R=
R+1:GOTO930
910 IFC=0THENGOTO790ELSEC=C-1:GOTO930
920 IFC=0ORR=0THENGOTO790ELSEC=C-1:R=R-
1
930 PUTSPRITE1,(C-3,R-4),C1:PUTSPRITE2,
(C-3,R-4),C2:GOTO790
940 'AGG. NUOVE COOR. CURSORE
950 'O FUN. Mx,y = LINE——
960 B$="BM":GOTO 980
970 B$="M":GOTO 1050
980 RR$=STR$(R):CC$=STR$(C):IF LEN(X$(Z

```

LIST

3

```

))>=150 OR H=1 THENGOTO 1000
990 X$(Z)=X$(Z)+B$+MID$(CC$,2,LEN(CC$))
+", "+MID$(RR$,2,LEN(RR$)):GOTO 1020
1000 IF H=0 THEN X$(Z)=X$(Z)+CHR$(34)
1010 Z=Z+1:NL=NL+1:X$(Z)=STR$(NL)+" DRA
W "+CHR$(34)+B$+MID$(CC$,2,LEN(CC$))+",
"+MID$(RR$,2,LEN(RR$))
1020 PLAY"F":P=0:K$="":J=0:W=0:V=0:L=0:
G=1:H=0:IF Z>=4 THENGOTO 1640 ELSE FORT
M=1TO500:NEXTTM:GOTO 400
1030 REM FUNZIONE LINE
1040 REM
1050 LINE(CO,RI)-(C,R),T:GOTO980
1060 REM AGG. FUN. LINE B o BF
1070 REM
1080 B$="B":LINE(CO,RI)-(C,R),T,B:GOTO
1100
1090 B$="BF":LINE(CO,RI)-(C,R),T,BF
1100 IF H=0 THEN X$(Z)=X$(Z)+CHR$(34)
1110 CO$=STR$(CO):RI$=STR$(RI):CC$=STR$
(C):RR$=STR$(R):T$=STR$(T):Z=Z+1:NL=NL+
1:H=1
1120 X$(Z)=STR$(NL)+" LINE("+MID$(CO$,2
,LEN(CO$))+", "+MID$(RI$,2,LEN(RI$))+")-
("+MID$(CC$,2,LEN(CC$))+", "+MID$(RR$,2,
LEN(RR$))+"), "+MID$(T$,2,LEN(T$))+", "+B
$
1130 GOTO 960
1140 REM AGG. FUN. CIRCLE
1150 REM
1160 IFH=0THEN X$(Z)=X$(Z)+CHR$(34)
1170 RC=ABS(CO-C):CIRCLE(CO,RI),RC,T
1180 RC$=STR$(RC):CO$=STR$(CO):RI$=STR$
(RI):T$=STR$(T):Z=Z+1:NL=NL+1:H=1
1190 X$(Z)=STR$(NL)+" CIRCLE("+MID$(CO$
,2,LEN(CO$))+", "+MID$(RI$,2,LEN(RI$))+
"), "+MID$(RC$,2,LEN(RC$))+", "+MID$(T$,2,
LEN(T$)):GOTO960
1200 REM AGG. FUN. DRAW
1210 REM
1220 IF G=1 THEN G=2:GOTO 1250
1230 IF LEN(X$(Z))>=150ORH=1THENGOTO129
0
1240 L$=STR$(L):X$(Z)=X$(Z)+F$+MID$(L$,
2,LEN(L$)):L=0:H=0
1250 ON V GOTO 480,500,520,540,560,580,
600,620
1260 REM INCR. NUM. DI LINEA
1270 REM E AGG. FUNZ. DRAW——
1280 REM
1290 PLAY"B":IFH=0THEN X$(Z)=X$(Z)+CHR$
(34)
1300 Z=Z+1:NL=NL+1:L$=STR$(L):X$(Z)=STR
$(NL)+CHR$(32)+"DRAW "+CHR$(34)+F$+MID$
(L$,2,LEN(L$)):L=0:H=0:IF Z>=4 THENGOTO
1640 ELSE 1250
1310 REM CAMBIO COLORE
1320 REM
1330 PLAY "A":T=T+1:IFT=16THENT=0
1340 T$=STR$(T):PUTSPRITE 0,(247,0),T:G
OTO 400
1350 REM AGG. COLORE
1360 REM
1370 IF LEN(X$(Z))>=150 OR H=1 THEN GOT
O 1420
1380 IF L=0 THEN1390ELSE1400
1390 PLAY"DB":X$(Z)=X$(Z)+"C"+MID$(T$,2
,LEN(T$)):GOTO 1410
1400 PLAY"DB":L$=STR$(L):X$(Z)=X$(Z)+F$
+MID$(L$,2,LEN(L$))+", "+MID$(T$,2,LEN(T

```


LIST

4

```

$)):GOTO 1410
1410 L=0:W=0:V=0:H=0:G=1:GOTO 400
1420 IFH=1ANDL=0THENGOTO1490
1430 IFH=1ANDL>0THENGOTO1510
1440 IFH=0ANDL=0THENGOTO1480ELSE1500
1450 REM INCR.NUM.LINEA E
1460 REM AGG. COLORE
1470 REM
1480 X$(Z)=X$(Z)+CHR$(34)
1490 PLAY"DB":Z=Z+1:NL=NL+1:X$(Z)=STR$(
NL)+" DRAW "+CHR$(34)+"C"+MID$(T$,2,LEN
(T$)):IF Z>=4 THENGOTO 1640 ELSE 1410
1500 X$(Z)=X$(Z)+CHR$(34)
1510 PLAY"DB":Z=Z+1:NL=NL+1:L$=STR$(L):
X$(Z)=STR$(NL)+" DRAW "+CHR$(34)+F$+MID
$(L$,2,LEN(L$))+ "C"+MID$(T$,2,LEN(T$)):
IF Z>=4 THENGOTO 1640 ELSE 1410
1520 REM AGG.FUN.PAINT
1530 REM
1540 PLAY"B":IF H=0ANDL=0 THENGOTO 1570
1550 IFH=0ANDL>0THENGOTO 1600
1560 IF H=1ANDL>0THENGOTO1610ELSE1580
1570 X$(Z)=X$(Z)+CHR$(34)
1580 PAINT (C,R),T:Z=Z+1:NL=NL+1:CC$=ST
R$(C):RR$=STR$(R):T$=STR$(T):X$(Z)=STR$(
NL)+" PAINT ("+MID$(CC$,2,LEN(CC$))+",
"+MID$(RR$,2,LEN(RR$))+")"+MID$(T$,2,LE
N(T$)):W=0:V=0:G=1:H=1:L=0
1590 IF Z>=4 THENGOTO 1640 ELSE 400
1600 L$=STR$(L):X$(Z)=X$(Z)+F$+MID$(L$,
2,LEN(L$)):GOTO 1570
1610 L$=STR$(L):Z=Z+1:NL=NL+1:X$(Z)=STR
$(NL)+" DRAW "+CHR$(34)+F$+MID$(L$,2,LE
N(L$)):GOTO 1570
1620 ' STAMPA AUTOPROGRAMMA
1630 '
1640 SCREEN 0:COLOR 15,4,4:CLS
1650 IF H=1 THENGOTO 1690
1660 IF L=0 THENGOTO 1680
1670 L$=STR$(L):X$(Z)=X$(Z)+F$+MID$(L$,
2,LEN(L$))+CHR$(34):GOTO 1710
1680 X$(Z)=X$(Z)+CHR$(34):GOTO1710
1690 IF L=0 THENGOTO 1710
1700 L$=STR$(L):Z=Z+1:NL=NL+1:X$(Z)=STR
$(NL)+" DRAW "+CHR$(34)+F$+MID$(L$,2,LE
N(L$))+CHR$(34):GOTO 1710
1710 CC$=STR$(C):RR$=STR$(R):NL=NL+1:NL
$=STR$(NL):T$=STR$(T)
1720 X$(8)="165 CLEAR1000:S$="+CHR$(34)
+S$+CHR$(34)+"P$="+CHR$(34)+P$+CHR$(34)
+"":NL$="+MID$(NL$,2,LEN(NL$))+":C$="+MID
$(CC$,2,LEN(CC$))+":R$="+MID$(RR$,2,LEN(
RR$))+":T$="+MID$(T$,2,LEN(T$))+":Y=1"
1730 LOCATE 0,0:PLAY"E"
1740 FOR S=0 TO Z
1750 PRINTX$(S)
1760 NEXT S
1770 PRINTX$(8)
1780 END
1790 'LISTATO AUTOPROGRAMMA
1800 ' (dalla linea 2000)
1810 '=====
10000 GOTO 165

```

ESEMPIO DI A.P.G. (Autoprogramma grafico) ELABORATO COMPLETAMENTE DAL COMPUTER.

```

10 SCREEN 2,0,0:COLOR 14,1,1:CLS
20 DRAW "BM100,100BM0,127R12F8R2"
30 PAINT (22,135),14
40 DRAW "R224BM0,127U6R12"
50 DRAW "BM12,121F8R226D6BM236,130"
60 PAINT (236,130),14
70 DRAW "R10U1C15U34L12U7L59D6L57U26L30
D18L31U7L39D6L18D33R7D1L4R9F8R226U9BM21
9,126"
80 PAINT (219,126),15
90 DRAW "BM219,126BM174,88C7D5L55U93R12
7D94L11U7L61D3BM201,86"
100 PAINT (201,86),7
110 DRAW "BM102,67C8U20R2L1D20L1R2U18D8
R1U1L2U2R2U5L1U1R1D2L1D1L2D4L1H1U1H1U1H
1U4BM104,51R2E1U1E1U3BM102,62L1D1L1D1L1
D3R2U3R2D1L3D1R5U2D1F1D1L7H1R5BM98,45"
120 DRAW "C2"
130 DRAW "BM98,45L1H1U1L2U2H1U2R1U1H1U1
R1U3R2U1R1E1U1E1U1E2U2E1R1F1R1F1R1F1D1F
1D1R1F1D1F1D1G1D1F2D3F1D1G1D1G1L1G1L1D1
G1U1H1L1G2L2H1L1H1L1H1L1BM104,36"
140 PAINT (104,36),2
150 DRAW "BM119,0C1D67U67BM176,87C4U65R
47BM231,87U65L50D65L5U11BM178,86"
160 PAINT (178,86),4
170 DRAW "BM178,86L2D7E5U5BM178,89"
180 PAINT (178,89),4
190 DRAW "BM231,94U16BM133,67C11R1"
200 LINE(134,67)-(158,32),11,BF
210 DRAW "BM158,32"
220 DRAW "BM158,32C13L24D36R25U36L26D36
R14L1R1L1R1U36L1D37L14D2R29U2L22R8L1U18
D1U1L12R25BM182,23"
230 LINE(182,23)-(230,87),13,BF
240 DRAW "BM230,87C11BM185,85R45BM186,2
5"
250 LINE(186,25)-(208,81),11,B
260 DRAW "BM208,81"
270 DRAW "BM208,81BM212,81U56D56BM216,8
1U56D56BM220,81U1D1U56C11BM212,28R8D3L8
D3R8D3L8D3R8D3L8D3R8D3U1D1L8D3R8D3L8D3R
8D3L8D3R8D3L8D3R8D3L8D3R8BM190,56"
280 DRAW "U1R1D1L1U1BM119,68C15D28BM84,
85C14U4R35D4L35U1R34U1L34U1R34BM87,68C8
D7U75D80L4D5L25U7L41D6L17U84R87D14BM46,
1"
290 PAINT (46,1),8
300 DRAW "BM39,30"
310 DRAW "BM39,30C14L1"
320 CIRCLE(38,30),13,10
330 DRAW "BM25,30R26D1C8"
340 LINE(51,31)-(22,47),8,BF
350 DRAW "BM22,47"
360 DRAW "BM22,47U13R3C14BM37,29"
370 PAINT (37,29),10
380 DRAW "BM25,33"
390 LINE(25,33)-(51,77),10,BF
400 DRAW "BM51,77"
410 DRAW "BM51,77BM63,38"
420 LINE(63,38)-(80,68),10,BF
430 DRAW "BM80,68BM13,68"
440 LINE(13,68)-(2,38),10,BF
450 DRAW "BM2,38"
460 DRAW "BM2,38C6BM62,37D32R19U69D1U1D
37L24D31R6L1U11BM60,55"

```

LIST ... la tua rivista


```

470 PAINT (60,55),6
480 DRAW "BM14,38D30U30BM81,38D47U2BM81
,83R6L31BM23,83L23D2R23D1L23R23BM56,86R
31C6BM73,0D1F1D2F3D4R1F2D2F2R1F1D2F1D1F
2BM74,3R14BM77,7R10BM80,11R7BM81,15"
490 DRAW "R6BM84,19"
500 DRAW "BM84,19BM78,3D4R7L1U4BM84,11D
4BM84,19R3BM88,11C1U11C9BM246,6L136U2E4
R132D6BM174,5"
510 PAINT (174,5),9
520 DRAW "L55R5C10E5D4R4E4D4R4E4D4R4E4D
4R4E4D4R4E4D4R4E4D4R4E4D4R4E4D4R4E4D4R4
E4D4R4E4D4R4E4D4R4E4D4R4E4D4R4BM95,117C
1R8U2H2U17R2U1H1U2H1L4G1D2L2D2R2"
530 DRAW "D17"
540 DRAW "BM96,113G2D2BM104,96D1U1D1L3B
M99,100"
550 PAINT (99,100),1
560 DRAW "BM98,98U1L4R10U1L4R4U1L3D6L2U
6R5D1L7R5U2R2D1L6BM101,96D18BM104,96C15
U5D11U10D1L2H3R9L6D1R6L8R8L6D1R5D2L4R6U
5L15D5L10U2R22U1L12R1BM97,92C1R1"
570 DRAW "L1"
580 DRAW "BM97,92R3U1L3BM95,98R1L1R7L4D
15L2D4L2R1D1R7L1D1L5BM95,116U20R4D3R2D1
7R1U20L2BM101,98C15D17BM99,99D16BM28,36
C1"
590 DRAW "BM28,36C8D1U1D4R1E1H1E1H1L1BM
34,40U1D1U3E1F1D3U1L1BM40,40U1D1U4R1F1G
1F1D1BM46,32D46U42D8U1L24BM25,32C1R1L1R
26BM25,78R14L14R26BM25,44"
600 DRAW "BM25,44C11R20D1L20D1R20D31L1U
30L1U1L1D31"
610 DRAW "BM40,48U1D30L2U30L2D30L2U30L2
D30L2U30"
620 DRAW "BM30,47R13L13D30R14BM87,87C14
L1R1L87"
630 DRAW "BM0,87C11BM64,54R4L4R16D2L16D
2R16BM38,24"
640 CIRCLE(38,24),3,11
650 DRAW "BM35,24BM38,24"
660 PAINT (38,24),11
670 DRAW "BM38,24BM92,3C10L92U3R88F3L14
BM77,2"
680 PAINT (77,2),10
690 DRAW "BM77,107"
700 GOTO 700

```



GIOCHI MATEMATICI

“CONVIVENZA PACIFICA”

n esclusiva per voi, lettori di LIST, un nuovo pas-
satempo di genere scacchistico, da giocare sia da
soli che in compagnia dei vostri amici.

L'idea di «Convivenza pacifica» è quella di porre un numero indeterminato di pezzi scacchistici in una normale scacchiera di 64 caselle, in modo che nessun pezzo possa, in una mossa, effettuare una qualsiasi presa. Il gioco può avere due varianti: l'una con lo spiegamento dei pezzi di un solo colore, che chiameremo versione «monocolore», l'altra con lo schieramento di pezzi di colore diverso, che chiameremo versione «bicolore».




Ognuna delle varianti al gioco ha delle precise regole a cui sottostare e dei punteggi per stabilire dei primati.

Il concetto di primato è insito nei giochi di ispirazione matematica: quale sarà il numero massimo di pezzi che si potranno immettere sulla scacchiera? O in alternativa, quale il punteggio massimo acquisibile?

Le illustrazioni seguenti non risolvono i due problemi, esse mostrano dei risultati parziali che richiedono di essere superati dai lettori.

Vediamo questi esempi:























PUNTI

	100
	90
	80
	70
	60
	50

♔				♗			♘
		♔					
♙				♙			♙
							♖
♖	♖		♖	♘			♖
					♗		
						♗	
♖	♘		♖	♖			♘

monocolore : 1250 P.

f. A

bicolore: 1320 P.

f. *B*

REGOLE NEL MONOCOLORE

1) Sulla scacchiera non possono essere presenti più di un RE e più di una REGINA, è permesso invece ampliarne la corte con altri Alfieri, Cavalli, Pedoni, Torri.

2) Si possono aggiungere pezzi supplementari solo dopo che siano stati disposti i 16 pezzi classici del gioco degli scacchi.

3) Almeno un alfiere deve poggiare su una casella di colore diverso dagli altri.

4) Il modo di mangiare dei pedoni si intende rivolto verso il sud della scacchiera.

REGOLE NEL BICOLORE

1) Non si possono aumentare i pezzi sulla scacchiera oltre ai 16 classici per ogni schieramento.

2) Si possono immettere altri pezzi sulla scacchiera solo quando siano in «convivenza pacifica» i RE e le REGINE di ogni schieramento.

3) Un alfiere dello stesso schieramento non può poggare su una casella del medesimo colore del suo paio.

4) Il modo di mangiare dei pedoni si intende rivolto a sud per i pezzi neri e a nord per i bianchi.

Conoscute le regole potete sbizzarrirvi nella ricerca dei record di ogni variante e se in casa non possedete delle scacchiere, carta e matita prenderanno il loro posto, marcando in una griglia 8x8 la lettera R per il RE, e la D per la REGINA, la T per la TORRE, la A per L'ALFIERE, la C per il CAVALLO E LA P per il PEDONE; nello stesso modo potrete inviare a LIST i primati raggiunti, i migliori dei quali saranno pubblicati con il nome dell'autore.

DIFFONDETE

“LIST”

**OGNI LETTORE
PROCURI
UN NUOVO LETTORE**

PIE CHART

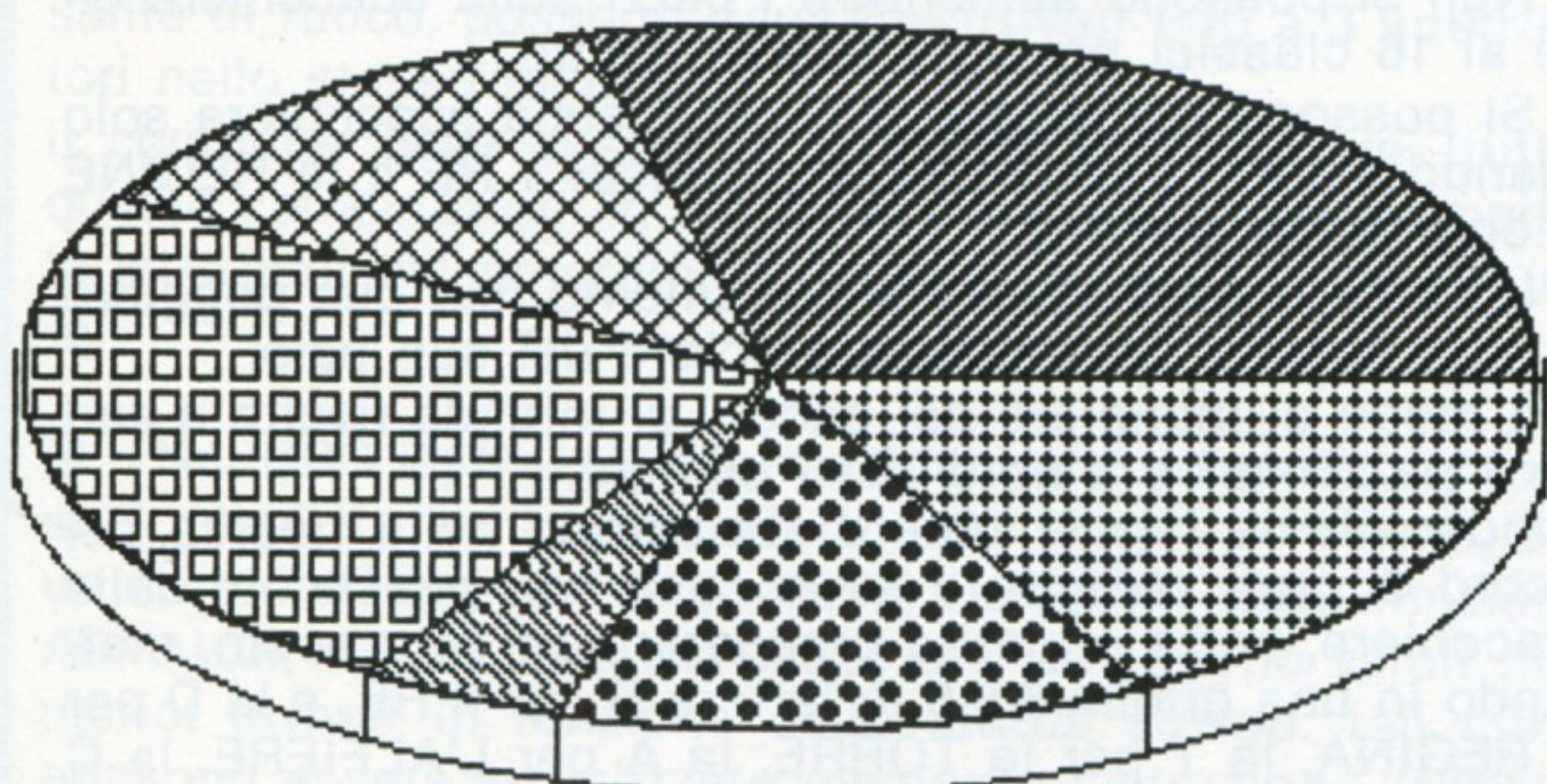


Grafici a torta per Olivetti Prodest.

Chissà quanti di voi, vedendo le varie trasmissioni sulle campagne elettorali primaverili, hanno ammirato le favolose proiezioni Doxa dove venivano rappresentati in un unico «tortino» tutti i partiti politici, a ciascuno dei quali veniva assegnata una «fetta» colorata differentemente dalle altre. Ebbene, perché accontentarsi di vedere questi famosi «Pie Charts» (così vengono chiamati dagli inglesi) soltanto in televisione? Perché non rappresentare, invece della percentuale dei verdi al Parlamento, anche la ripartizione delle spese condominiali, o le entrate annuali mese per mese? Ancora una volta il piccolo mostriciattolo grigio può aiutarci a risolvere il problema: ecco dunque, per la gioia dei vostri occhi, un elegantissimo programma per diagrammi a torta, creato sfruttando al massimo le notevoli possibilità grafiche del piccolo Prodest.

Prima di passare alle spiegazioni sull'uso del programma, raccomandiamo ancora una volta ai lettori di fare molta attenzione nella digitazione del listato; rispettate anche gli spazi nelle righe di «PRINT», e badate ai nomi delle variabili, spesso infidi e traditori!!! Se comunque non volete affaticare troppo i polpastrelli, vi ricordiamo che il programma è disponibile in redazione.

L'uso di PIE CHART è molto semplice e alla portata di tutti, neofiti e non. Le opzioni disponibili appena dato il RUN sono 5, richiamabili con la sola pressione del tasto corrispondente. Come al solito, analizziamole in sequenza.



⌘ GEN 16.8% ⌘ FEB 12.7% ⌘ MAR 4.5%
⌘ APR 24.3% ⌘ MAG 12.7% ⌘ GIU 29.1%

1 - Inserimento dati

Permette di inserire i valori da visualizzare (massimo 15) con le loro rispettive etichette. Un'etichetta è una parola o una lettera associata al valore in questione, e ne permette l'identificazione (ad esempio: GEN 19.87, FEB 34.5, ecc.). Per scegliere il valore da inserire fate spostare il simbolo ">" con i tasti di cursore e premete la barra spaziatrice. A questo punto verranno richieste, in sequenza, l'etichetta ed il valore associato. Le richieste sono normali INPUT, quindi battete ENTER per inserire il dato. Avvertiamo che i dati verranno visualizzati in sequenza dal primo all'ultimo, dove per ultimo si intende l'elemento più in basso. Se avete inserito i primi 3 elementi e passate direttamente al settimo, il quarto il quinto ed il sesto verranno considerati come elementi di valore 0 e senza etichetta.

Quando avete terminato l'inserimento, premete il tasto STOP per tornare al menu principale.

2 - Cambio PATTERN

Questa è una interessante opzione che vi permette di cambiare il colore ed il modello di riempimento di ciascuno degli elementi inseriti. Potete cioè riempire lo «spicchio» con quello che volete: lettere, numeri, colore pieno, ecc. All'inizio i riempimenti sono definiti dal computer con numeri e lettere. Per alterare il modello di un elemento dovete posizionarvi con i tasti cursore su è giù su quello desiderato, scegliere il carattere di riempimento con i tasti cursore destra e sinistra (sono disponibili una quindicina di caratteri programmabili predefiniti, più i numeri, le lettere e i vari simboli presenti sulla tastiera) e premere lo spazio per selezionare. Accanto all'elemento corrispondente verrà stampata una serie di caratteri come quello scelto, mentre in basso a destra un piccolo rettangolo si riempirà con quel modello, per rendersi conto del tipo di disegno che si otterrà. Potete anche cambiare il colore del modello, scegliendo tra i tre disponibili. Usate i tasti "+" e "-". Come conferma dell'operazione avvenuta, la stringa dei caratteri selezionabili cambierà colore. Per tornare al menu principale premete STOP.

3 - Esecuzione diagramma

Crea il diagramma basandosi sui dati precedentemente inseriti. Una volta terminato il disegno, gli elementi verranno visualizzati in fondo allo schermo secondo il formato selezionabile dal menu principale premendo la barra spaziatrice: SOLO ETICHETTA, ETICHETTA + VALORE, ETICHETTA + PERCENTUALE, SOLO VALORE. A seconda del formato scelto potranno essere visualizzati più o meno caratteri e numeri. Il massimo consentito per le etichette, in ogni caso, è di 10 caratteri.

Alla fine della visualizzazione potete ottenere l'output su stampante (DM 90) della videata, semplicemente premendo il tasto ACC. Per tornare al menu principale un tasto qualsiasi.

4 - Inizializza variabili

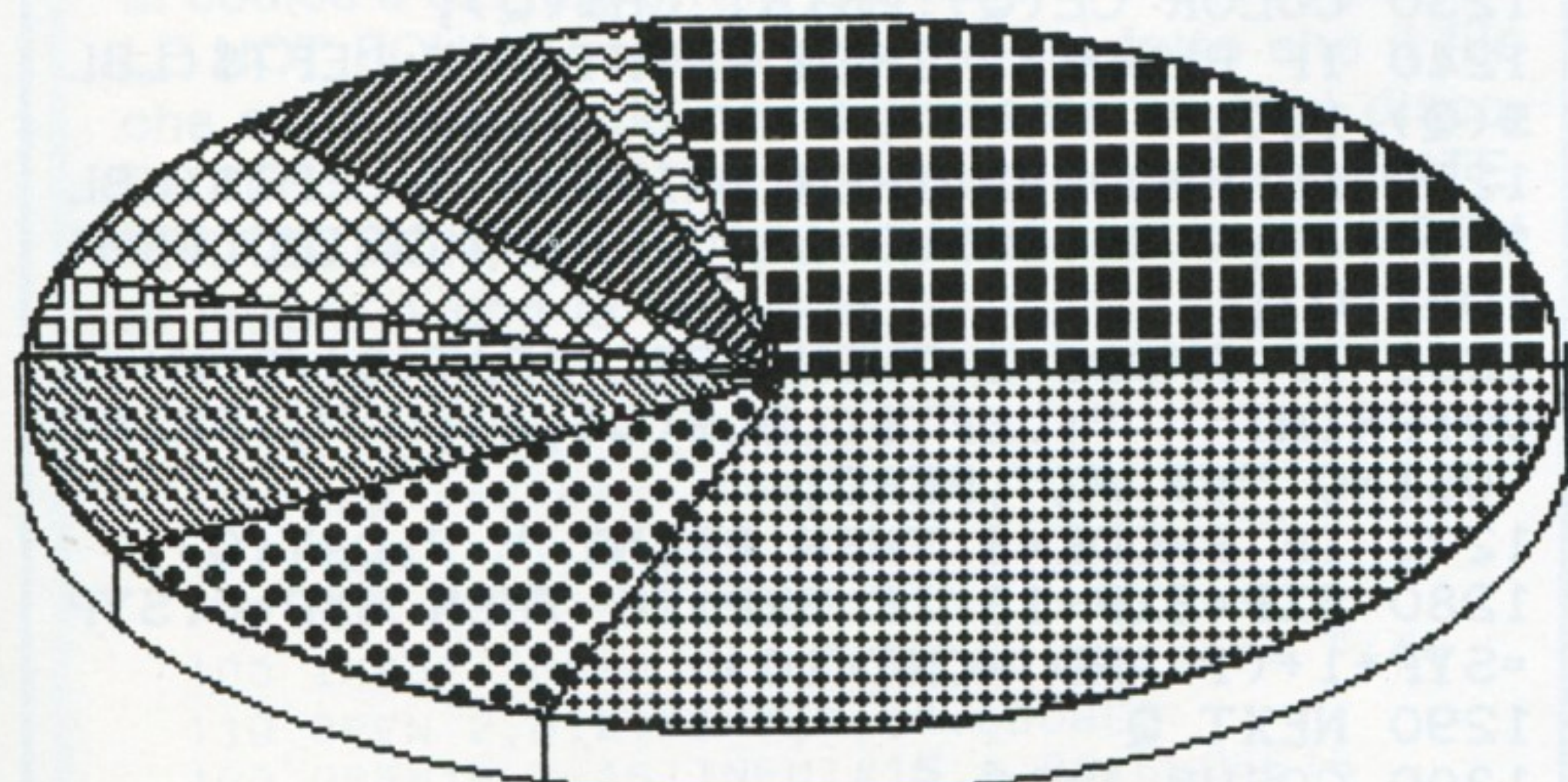
Cancella tutti i dati presenti in memoria, compresi i patterns. Ricordiamo che il numero dell'elemento più basso inserito tramite l'opzione 1 è cancellabile solo mediante questo comando.

5 - Fine lavoro

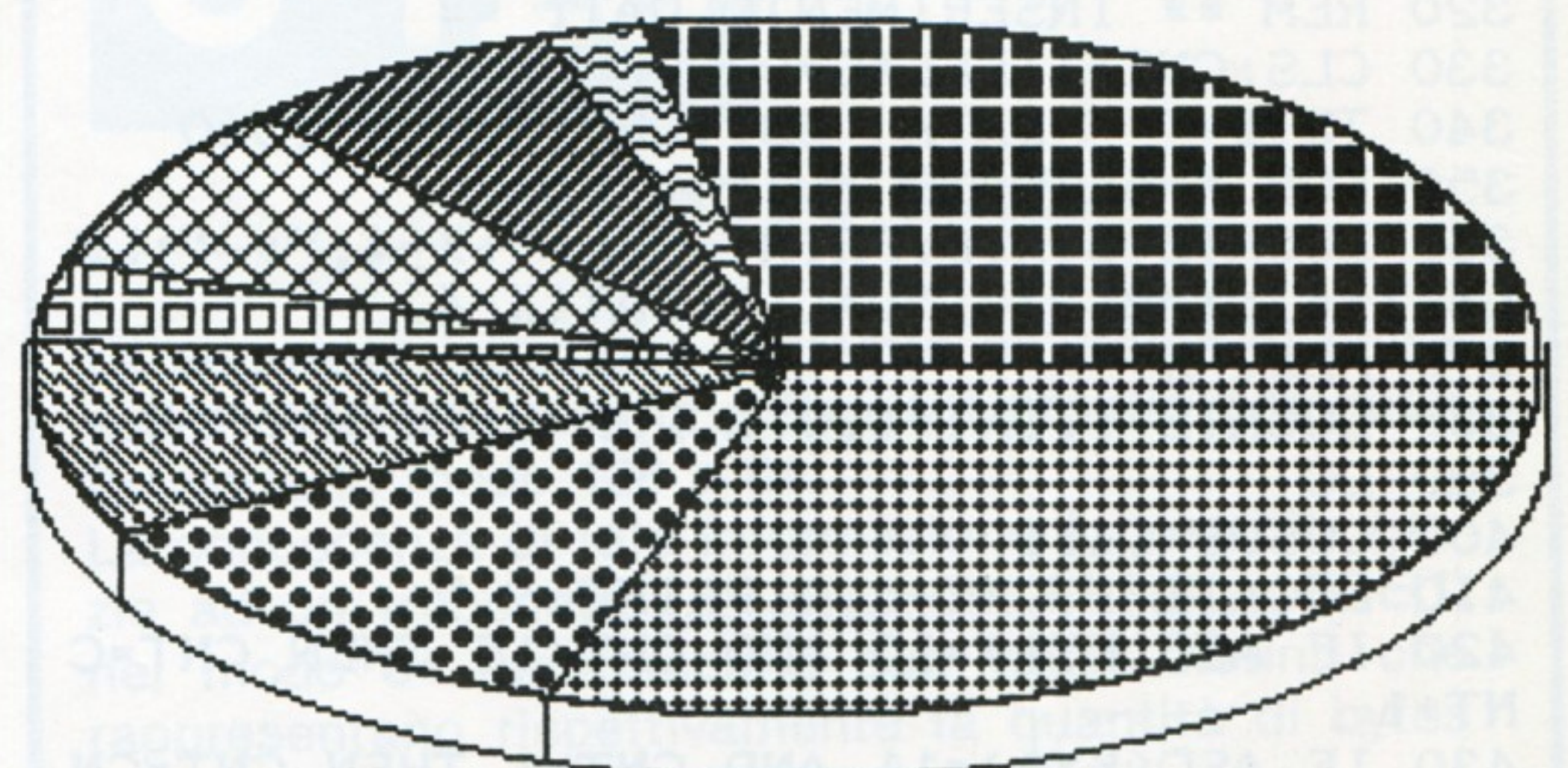
Ferma il programma, ritornando al BASIC 128.

Come avete visto, l'uso di PIE CHART è veramente immediato. Il programma è come al solito «smanettabilissimo», in quanto costituito esclusivamente da subroutines «farcite» di REM (visto che parliamo di torte, «farcite» è proprio la parola adatta!!!). Riguardo ai caratteri programmabili predefiniti, potete alterarli a piacimento, così come potete variare contenuto e lunghezza della stringa, definita alla linea 40 e contenente i patterns disponibili. Le variabili principali sono definite alla linea 50 (i vettori sono dimensionati a 16 anziché a 15) e sono: DAT (valori numerici), LBL\$ (etichette), CH\$ (patterns), CL\$ (colori associati) e PMOD\$ (modo di visualizzazione valori).

Il programma si presta molto a migliorie e perfezionamenti, che non abbiamo inserito per ragioni di tempo e di spazio. Vi esortiamo ancora una volta ad inviarci i vostri consigli ed anche, perché no, i programmi modificati. Buon lavoro, dunque, ma prima di tutto buon divertimento con PIE CHART!



INGE 30.2% EC.C 11.9% GIUR 9.1%
MAGI 3.7% MFN 7.8% ARCH 7.2%
FILO 2.1% BOH? 28.1%



INGE 30.2% EC.C 11.9% GIUR 9.1%
MAGI 3.7% MFN 7.8% ARCH 7.2%
FILO 2.1% BOH? 28.1%

LIST

1

```

1 REM *****
2 REM *   PIE CHART V 1.0   *
3 REM *   BY BETASOFT     *
4 REM *   GIULIO VANNINI  *
5 REM *   ANGELO COCCETTINI *
6 REM *   COPYRIGHT LIST 1987 *
7 REM *****
8 REM
10 CONSOLE ,,,2:SCREEN , ,0:CLEAR , ,16
20 GOSUB 1640
30 PAT$="" :FOR Q=0 TO 15: PAT$=PAT$+GR$(
Q):NEXT Q
40 PAT$=PAT$+"1234567890ABCDEFGHIJKLMN
OPQRSTUVWXYZ[ ]!$%&/'()*=?`@+*^,;.:_->< "
50 DIM DAT(16),LBL$(16),CH$(16),CL$(16),P
MOD$(4):ESCI=0
60 GOSUB 1480
70 PMOD$(1)="SOLO ETICHETTA":PMOD$(2)="
ETICHETTA + VAL":PMOD$(3)="ETICHETTA + %
":PMOD$(4)="SOLO VALORE":PMODE=1
80 DO
90 CLS
100 TST$="P I E   C H A R T":GOSUB 1540
110 BOX(0,0)-(319,17),2
120 LOCATE 0,8,0
130 PRINT "1 - INSERIMENTO DATI"
140 PRINT "2 - CAMBIO PATTERN"
150 PRINT "3 - ESECUZIONE DIAGRAMMA"
160 PRINT "4 - INIZIALIZZA VARIABILI"
170 PRINT "5 - FINE LAVORO"
180 PRINT:PRINT "  MODO VISUALIZZAZIONE:
":;:COLOR 3:PRINT PMOD$(PMODE)
190 LOCATE 0,18,0:PRINT "SELEZIONA: 1,2,
3,4 0 (SPACE) (MODO VIS.)"
200 GOSUB 1410
210 IF ASC(KY$)=32 THEN PMODE=PMODE+1:IF
PMODE>4 THEN PMODE=1
220 LOCATE 24,14:PRINT PMOD$(PMODE)
230 IF ASC(KY$)<49 OR ASC(KY$)>53 THEN G
OTO 200
240 N=ASC(KY$)-48
250 ON N GOSUB 320,590,980,1440,1590
260 IF ESCI THEN EXIT
270 LOOP
280 STOP
290 REM
    
```


LIST

4

```

1340 NUMBER$=STR$(DAT(Q)):FRM$="#####"
1350 IF MID$(NUMBER$,2,1)="." THEN NUMBE
R$="?" + NUMBER$
1360 FOR LP=2 TO LEN(NUMBER$)
1370 IF MID$(NUMBER$,LP,1)="." THEN LP=L
P-1:GOTO 1390
1380 NEXT LP
1390 IF LP<7 THEN MID$(FRM$,LP,1)="."
1400 RETURN
1410 REM ** ATTESA TASTO **
1420 KY$=INKEY$:IF KY$="" THEN GOTO 1420
1430 RETURN
1440 REM ** INIZIALIZZA VARIABILI **
1450 CLS:LOCATE 0,22
1460 PRINT " CONFERMI ? (S/N)"
1470 GOSUB 1410:IF KY$<>"S" THEN RETURN
1480 FOR Q=1 TO 16
1490 DAT(Q)=0:LBL$(Q)=" "
1500 NEXT Q:MCNT=0
1510 FOR Q=1 TO 15 STEP 3:CL(Q)=1:CL(Q+1
)=2:CL(Q+2)=3:NEXT Q:CL(16)=1
1520 FOR Q=1 TO 15:CH$(Q)=MID$(PAT$,Q+17
,1):NEXT Q
1530 RETURN
1540 REM ** DISEGNA TESTATA **
1550 ATTRB 1,1:COLOR 1,0:LOCATE 2,1
1560 PRINT TST$:ATTRB 0,0:COLOR 2,0
1570 BOX(0,0)-(319,17),2
1580 RETURN
1590 REM ** FINE LAVORO **
1600 CLS:LOCATE 0,22

```

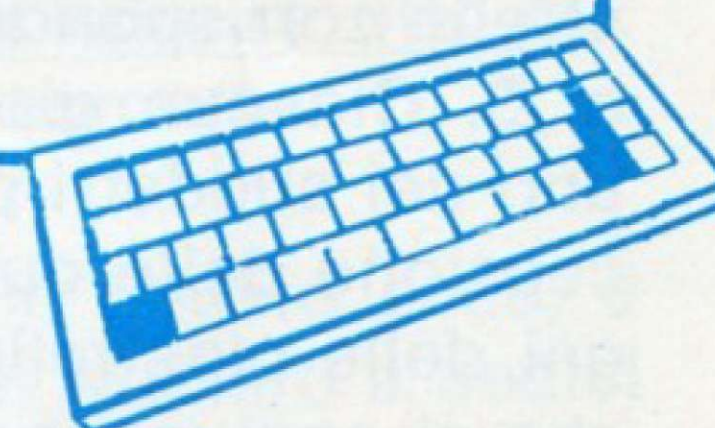
LIST

5

```

1610 PRINT " CONFERMI ? (S/N)"
1620 GOSUB 1410:IF KY$<>"S" THEN RETURN
1630 ESCI=1:RETURN
1640 REM ** DEFINISCE CARATTERI **
1650 DEFGR$(0)=255,255,255,255,255,255,2
55,255
1660 DEFGR$(1)=34,119,34,0,34,119,34,0
1670 DEFGR$(2)=195,129,24,60,60,24,129,1
95
1680 DEFGR$(3)=145,72,36,146,73,36,18,13
7
1690 DEFGR$(4)=0,60,66,129,60,66,129,0
1700 DEFGR$(5)=36,36,231,0,0,231,36,36
1710 DEFGR$(6)=129,66,36,24,24,36,66,129
1720 DEFGR$(7)=16,16,8,196,35,16,8,8
1730 DEFGR$(8)=231,129,165,24,24,165,129
,231
1740 DEFGR$(9)=36,24,129,66,66,129,24,36
1750 DEFGR$(10)=231,153,153,126,126,153,
153,231
1760 DEFGR$(11)=66,165,66,0,0,66,165,66
1770 DEFGR$(12)=153,36,36,153,153,36,36,
153
1780 DEFGR$(13)=102,102,102,102,102,102,
102,102
1790 DEFGR$(14)=102,204,153,51,102,204,1
53,51
1800 DEFGR$(15)=231,231,231,0,0,231,231,
231
1810 RETURN

```



PER I LETTORI "AUTORI"

INVITO A COLLABORARE

I lettori-autori, i quali abbiano elaborato programmi originali di loro creazione e di interesse comune o si sentano capaci di crearne e desiderino vederli pubblicati, possono inviarli — purché registrati su supporto magnetico — alla Redazione di "LIST", Via Flavio Stilicone, 111, Cap. 00175, Roma, tel. 7665094.

Gli elaborati devono essere corredati di chiare note esplicative sul funzionamento ed applicazione del programma.

Gli Autori si assumono ogni responsabilità sull'originalità dei loro elaborati.

I programmi accettati e pubblicati saranno compensati in ragione della loro originalità, chiarezza e precisione di elaborazione. Saranno preferiti programmi brevi e di facile esecuzione.

Prima di inviare i programmi è preferibile telefonare a «LIST» preannunciando l'invio.

"LIST"

LIST

2

```

300 REM ** INIZIO SUBROUTINES **
310 REM
320 REM ** INSERIMENTO DATI **
330 CLS: CNT=1: EFLAG=0
340 TST$=" DATA EDITOR": GOSUB 1540
350 FOR X=5 TO 19
360 LOCATE 3,X,0: PRINT "EL."; X-4; ": "; LBL$(X-4); TAB(23); "VAL: "; DAT(X-4)
370 NEXT X
380 LOCATE 1, CNT+4, 0: COLOR 2: PRINT ">"
390 DO
400 GOSUB 1410
410 LOCATE 1, CNT+4, 0: PRINT " "
420 IF ASC(KY$)=10 AND CNT<15 THEN CNT=CN
NT+1
430 IF ASC(KY$)=11 AND CNT>1 THEN CNT=CN
T-1
440 IF ASC(KY$)=32 THEN EXIT
450 IF ASC(KY$)=2 THEN EFLAG=1: EXIT
460 LOCATE 1, CNT+4, 0: PRINT ">"
470 LOOP
480 IF EFLAG=1 THEN RETURN
490 IF CNT>MCNT THEN MCNT=CN
T
500 CONSOLE CNT+4, CNT+4
510 LOCATE 0, CNT+4, 0: PRINT "
"
520 LOCATE 3, CNT+4, 0: PRINT "EL."; CNT; ":
";: INPUT LBL$(CNT)
530 IF LEN(LBL$(CNT))>10 THEN GOTO 510
540 LOCATE 3, CNT+4, 0: PRINT "EL."; CNT; ":
"; LBL$(CNT); TAB(23); "VAL: ";
550 LOCATE 27, CNT+4, 0: INPUT DAT(CNT)
560 CONSOLE 0, 24
570 LOCATE 3, CNT+4, 0: PRINT "EL."; CNT; ":
"; LBL$(CNT); TAB(23); "VAL: "; DAT(CNT)
580 GOTO 380
590 REM ** CAMBIO PATTERN ***
600 CLS: CNT=1: PS=2: PCOL=1: PATTERN GR$(0)
: EFLAG=0
610 TST$=" PATTERN EDITOR": GOSUB 1540
620 FOR X=3 TO 17
630 LOCATE 3,X,0: PRINT "EL."; X-2; ": "; LBL$(X-2); TAB(23);
640 COLOR CL(X-2): FOR Q=1 TO 6: PRINT CH$(X-2);: NEXT Q
650 COLOR 3: NEXT X
660 ATTRB 1,1: COLOR 1: LOCATE 7,21,0: PRIN
T MID$(PAT$,2,1): LOCATE 10,21,0: PRINT MI
D$(PAT$,3,1)
670 ATTRB 1,0: LOCATE 7,22,0: COLOR 3: PRIN
T " "": ATTRB 0,0
680 BOXF(250,140)-(300,190),1
690 LOCATE 1, CNT+2, 0: COLOR 2: PRINT ">"
700 DO
710 GOSUB 1410
720 LOCATE 1, CNT+2, 0: PRINT " "
730 IF ASC(KY$)=10 AND CNT<15 THEN CNT=CN
NT+1
740 IF ASC(KY$)=11 AND CNT>1 THEN CNT=CN
T-1
750 IF ASC(KY$)=8 AND PS>2 THEN PS=PS-1
760 IF ASC(KY$)=9 AND PS<(LEN(PAT$)-1) T
HEN PS=PS+1
770 IF ASC(KY$)=43 THEN PCOL=PCOL+1: IF P
COL>3 THEN PCOL=1
780 IF ASC(KY$)=45 THEN PCOL=PCOL-1: IF P
COL<1 THEN PCOL=3
790 IF ASC(KY$)=32 THEN EXIT
800 IF ASC(KY$)=2 THEN EFLAG=1: EXIT

```

LIST

3

```

810 LOCATE 1, CNT+2, 0: PRINT ">"
820 GOSUB 910
830 LOOP
840 IF EFLAG=1 THEN RETURN
850 PATTERN MID$(PAT$,PS,1)
860 BOXF(250,140)-(300,190),PCOL
870 COLOR PCOL: LOCATE 23, CNT+2
880 FOR Q=1 TO 6: PRINT MID$(PAT$,PS,1);:
NEXT Q
890 CH$(CNT)=MID$(PAT$,PS,1): CL(CNT)=PCO
L
900 GOTO 690
910 REM ** STAMPA CARATTERI **
920 LOCATE 4,21,0: COLOR PCOL
930 FOR W=1 TO 3
940 ATTRB 1,1: PRINT MID$(PAT$,PS-2+W,1);
: ATTRB 0,0
950 PRINT " ";
960 NEXT W
970 COLOR 3: RETURN
980 REM ** ESEGUE DIAGRAMMA **
990 CLS
1000 CIRCLE(160,85)151,71,3
1010 FOR CIL=85 TO 72 STEP-1
1020 CIRCLE(160,CIL)150,70,5
1030 NEXT CIL
1040 CIRCLEF(160,73)149,70,0
1050 LINE(311,69)-(311,96): LINE(9,69)-
(9,96)
1060 REM GOTO 2149
1070 SUM=0: CL=0: AST=0
1080 FOR Q=1 TO MCNT: SUM=SUM+DAT(Q): NEXT
Q
1090 FOR Q=1 TO MCNT
1100 AFN=AST+(6.28318*(DAT(Q)/SUM))
1110 PATTERN CH$(Q)
1120 SXP=160+150*COS(AST): SYP=73+71*SIN(
AST)
1130 FXP=160+150*COS(AFN): FYP=73+71*SIN(
AFN)
1140 LINE(160,73)-(SXP,SYP),3: LINE(160
,73)-(FXP,FYP),3
1150 IF AFN>=0.1 AND AFN<=3.13 THEN LINE
(FXP,FYP)-(FXP,FYP+13),3
1160 APT=AST+((AFN-AST)/2): PAINT(160+75*
COS(APT),73+35*SIN(APT)),CL(Q)
1170 AST=AFN
1180 NEXT Q
1190 CIRCLE(160,73)149,70,3
1200 SXP=0: SYP=20
1210 FOR Q=1 TO MCNT
1220 LOCATE SXP,SYP,0
1230 COLOR CL(Q): PRINT CH$(Q);
1240 IF PMODE=1 THEN PRINT " "; LEFT$(LBL
$(Q),10);
1250 IF PMODE=2 THEN PRINT " "; LEFT$(LBL
$(Q),4); " ";: GOSUB 1330: PRINT USING FRM$
; DAT(Q);
1260 IF PMODE=3 THEN PERC=(DAT(Q)/SUM)*1
00: PRINT " "; LEFT$(LBL$(Q),4); " ";: PRINT
USING "###.##%"; PERC;
1270 IF PMODE=4 THEN PRINT " "; DAT(Q);
1280 SXP=SXP+13: IF SXP>26 THEN SXP=0: SYP
=SYP+1+(1 AND MCNT<10)
1290 NEXT Q
1300 GOSUB 1410
1310 IF ASC(KY$)=22 THEN SCREENPRINT
1320 RETURN
1330 REM ** FORMATO NUMERO **

```




Tattica e colpo d'occhio per questo, rompicapo da tavolo, o meglio... da monitor per il vostro olivetti PC 128

Squiggle è un programma che non mancherà di incontrare il favore degli appassionati di giochi tipo «Nim». Lo potremmo definire come un lontano... parente del famoso domino. Si gioca in due su una plancia di 8x8 caselle. Ogni giocatore ne ha a disposizione un numero illimitato di «piastrelle» il cui disegno è fatto in tre modi diversi: a croce, a doppio quarto di cerchio, a doppio quarto sfasato di novanta gradi. Ogni tipo di piastrelle corrisponde ad un determinato codice di pezzo (X, Y o Z). Ciascun giocatore, a turno, deve piazzare a scelta uno dei suddetti pezzi, partendo in alto a sinistra. La successione delle curve e dei rettilinei determinata dai disegni delle piastrelle crea una specie di «percorso». Via via che si aggiungono a turno i pezzi si possono creare degli «intrecci», cioè la linea può ritorcersi su un pezzo precedentemente piazzato a continuare attraverso di esso in un'altra direzione. Scopo del gioco è arrivare per primi nell'angolo in basso a destra della griglia di gioco, oppure

forzare l'avversario ad uscire dalla scacchiera. Senza avere mai visto il gioco questo discorso potrà sembrare incomprendibile, ma se digiterete questo breve programma vi accorgete di quanto le regole siano banali.

Anche se a prima vista vi sembrerà un gioco stupido, provate a fare qualche partita completa e vedrete che SQUIGGLE richiede tattica, abilità e soprattutto colpo d'occhio per non perdere la... strada. Infatti il computer segue automaticamente il percorso, ma voi dovete ricordarvi dove andrà messa la pedina successiva.

Per quanto il funzionamento del programma, non ci dovrebbero essere problemi: all'inizio vengono richiesti i nomi dei due giocatori; per scegliere un pezzo bisogna solo premere il tasto corrispondente alla lettera a fianco di ciascuno di essi. Errori e fine partita vengono identificati automaticamente. Il computer terrà inoltre il conto dei numeri di partite vinte da ciascun giocatore, nel caso essi continuino a giocare.

LIST

1

```

1 REM *****
2 REM *      SQUIGGLE      PC 128      *
3 REM *  BY BETASOFT - LIST 1987  *
4 REM *  G. VANNINI - A. COCCETTINI *
5 REM *****
6 REM
10 SCREEN 7,4,4:CLS
20 CLEAR,,13:GOSUB 1110
30 GOSUB 1030:GOSUB 930
40 CLS
50 MVES=0:AX=1:AY=1:X=1:Y=3:X1=1:Y1=1:PL
R=START
60 FOR A=1 TO 8
70 FOR B=1 TO 8
80 A(A,B)=0:NEXT B
90 NEXT A:A(1,1)=1
100 GOSUB 810
110 LOCATE 29,5,0:COLOR 2:PRINT "
":LOCATE 29,5:PRINT P$(PLR)
120 COLOR 7:BEEP
130 Z$=INKEY$:IF Z$="" THEN GOTO 130
140 IF Z$<"X" OR Z$>"Z" THEN GOTO 130
150 COLOR 2
160 IF Z$="X" THEN LOCATE Y,X,0:PRINT A$
(1):LOCATE Y,X+1,0:PRINT A$(2):A(AX,AY)=
1
170 IF Z$="Y" THEN LOCATE Y,X,0:PRINT B$

```

LIST

2

```

(1):LOCATE Y,X+1,0:PRINT B$(2):A(AX,AY)=
2
180 IF Z$="Z" THEN LOCATE Y,X,0:PRINT C$
(1):LOCATE Y,X+1,0:PRINT C$(2):A(AX,AY)=
3
190 COLOR 7
200 REM *** CONTROLLA MOSSA ***
210 PLAY "DO"
220 IF A(AX,AY)=1 THEN GOTO 250
230 IF A(AX,AY)=2 THEN GOTO 290
240 IF A(AX,AY)=3 THEN GOTO 330
250 IF X1<X THEN X1=X:Y1=Y:AX=AX+1:X=X+2
:GOTO 370
260 IF X1>X THEN X1=X:Y1=Y:AX=AX-1:X=X-2
:GOTO 370
270 IF Y1<Y THEN X1=X:Y1=Y:AY=AY+1:Y=Y+2
:GOTO 370
280 IF Y1>Y THEN X1=X:Y1=Y:AY=AY-1:Y=Y-2
:GOTO 370
290 IF X1<X THEN X1=X:Y1=Y:AY=AY-1:Y=Y-2
:GOTO 370
300 IF X1>X THEN X1=X:Y1=Y:AY=AY+1:Y=Y+2
:GOTO 370
310 IF Y1<Y THEN X1=X:Y1=Y:AX=AX-1:X=X-2
:GOTO 370
320 IF Y1>Y THEN X1=X:Y1=Y:AX=AX+1:X=X+2
:GOTO 370

```


LIST

3

```

330 IF X1<X THEN X1=X:Y1=Y:AY=AY+1:Y=Y+2
:GOTO 370
340 IF X1>X THEN X1=X:Y1=Y:AY=AY-1:Y=Y-2
:GOTO 370
350 IF Y1<Y THEN X1=X:Y1=Y:AX=AX+1:X=X+2
:GOTO 370
360 IF Y1>Y THEN X1=X:Y1=Y:AX=AX-1:X=X-2
:GOTO 370
370 IF X<1 OR X>16 OR Y<3 OR Y>18 THEN W
IN=1:PLAY "L2OMIMIMIL50DD":GOTO 440
380 IF X=15 AND Y=17 THEN WIN=2:GOTO 440
390 IF A(AX,AY)<>0 THEN GOTO 220
400 MVES=MVES+1:PLR=PLR+1
410 IF PLR>2 THEN PLR=1
420 COLOR 5:LOCATE 28,8:PRINT MVES :COLO
R 7
430 GOTO 110
440 REM *** FINE GIOCO ***
450 CLS:GOSUB 680
460 LOCATE 0,14,0
470 IF WIN=1 THEN COLOR 2:PRINT P$(PLR);
:COLOR 7:PRINT "perde la partita":PRINT
"uscendo dal tavolo di gioco..."
480 IF WIN=2 THEN GOTO 500
490 GOTO 520
500 PLR=PLR+1:IF PLR>2 THEN PLR=1
510 COLOR 2:PRINT P$(PLR);:COLOR 7:PRINT
"vince in ";MVES;" mosse."
520 LOCATE 0,17,0:IF WIN=1 THEN PLR=PLR+
1:IF PLR>2 THEN PLR=1
530 S(PLR)=S(PLR)+1
540 COLOR 5:PRINT "PARTITE VINTE FINO AD
ORA: ":PRINT
550 COLOR 10:PRINT P$(1);:COLOR 7:PRINT
S(1)
560 COLOR 10:PRINT P$(2);:COLOR 7:PRINT
S(2)
570 LOCATE 0,22,0:PRINT "Vuoi giocare an
cora ? (S/N)"
580 Z$=INKEY$:IF Z$="" THEN GOTO 580
590 IF Z$="N" THEN STOP
600 LOCATE 0,22,0:PRINT "I giocatori son
o gli stessi ? (S/N) "
610 Z$=INKEY$:IF Z$="" THEN GOTO 610
620 IF Z$="S" THEN GOTO 640
630 RUN
640 START=START+1:IF START>2 THEN START=
1
650 LOCATE 0,23:PRINT "OK... Inizia ";P$
(START)
660 FOR A=1 TO 1000:NEXT A
670 CLS:GOTO 50
680 REM *** DISEGNA TITOLO ***
690 RESTORE 740
700 READ A:READ B:IF A=255 THEN COLOR ,4
:RETURN
710 LOCATE B+3,A+1,0:COLOR ,0:PRINT " "
720 LOCATE B+2,A,0:COLOR ,2:PRINT " "
730 GOTO 700
740 DATA 6,0,6,1,6,2,7,0,8,0,8,1,8,2,9,2
,10,0,10,1,10,2,7,4,7,5,7,6,8,4,8,6,9,4,
9,5,9,6,10,6,11,6,12,6,12,7,7,8,7,10,8,8
,8,10,9,8,9,9,9,10
750 DATA 5,12,7,12,8,12,9,12,7,14,7,15,7
,16,8,14,8,16,9,14,9,15,9,16,10,16,11,14
,11,15,11,16,7,18,7,19,7,20,8,18,8,20,9,
18,9,19,9,20,10,20,11,18,11,19,11,20
760 DATA 5,22,6,22,7,22,8,22,9,22,9,23,6
,25,6,26,6,27,7,25,8,25,8,26,8,27,9,25,1
0,25,10,26,10,27,255,255
770 REM *** TITOLO ***

```

LIST

4

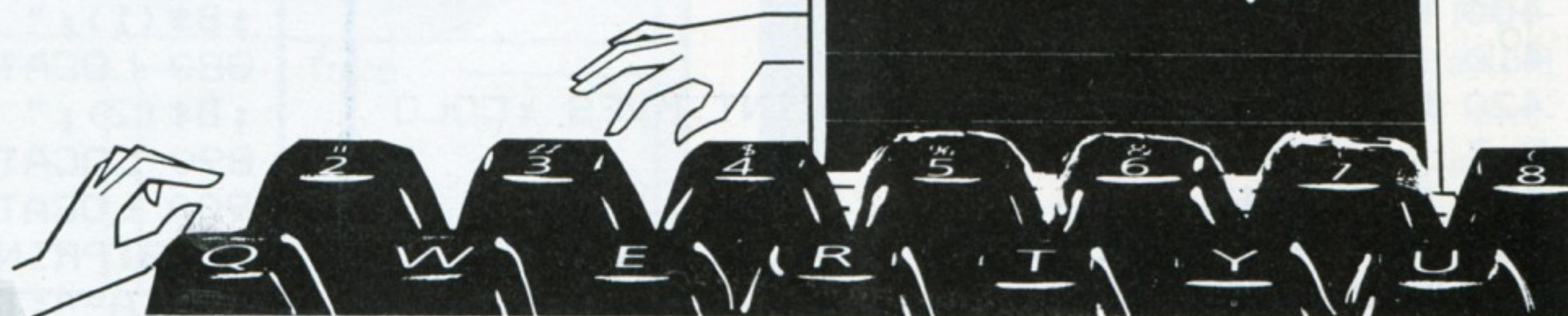
```

780 ATTRB 1,1:COLOR C:LOCATE A,B:PRINT "
S Q U I G G L E"
790 ATTRB 0,0:COLOR 7
800 RETURN
810 REM *** DISEGNA TAVOLO ***
820 BOX (0,0)-(319,199),9
830 FOR A=3 TO 17 STEP 2
840 FOR B=1 TO 15 STEP 2
850 LOCATE A,B,0:PRINT D$(1):LOCATE A,B+
1,0:PRINT D$(2)
860 NEXT B:NEXT A
870 LOCATE 20,1,0:PRINT "X ";A$(1);" Y "
;B$(1);" Z ";C$(1)
880 LOCATE 20,2,0:PRINT " ";A$(2);" "
;B$(2);" ";C$(2)
890 LOCATE 21,5,0:PRINT "MUOVE:"
900 LOCATE 21,8,0:PRINT "TOTALE":LOCATE
21,8:PRINT "MOSSE: 0"
910 A=4:B=19:C=1:GOSUB 770
920 RETURN
930 REM *** INIZIO GIOCO ***
940 GOSUB 680
950 LOCATE 0,15,0:INPUT "GIOCATORE 1: ";
P$(1)
960 INPUT "GIOCATORE 2: ";P$(2)
970 FOR A=1 TO 2:IF LEN (P$(A))>10 THEN
P$(A)=LEFT$(P$(A),10)
980 NEXT A
990 LOCATE 0,20,0:PRINT "Premi un tasto
per iniziare."
1000 PRINT:PRINT P$(START);" inizia per
primo..."
1010 IF INKEY$="" THEN GOTO 1010
1020 RETURN
1030 REM *** VARIABILI ***
1040 DIM A$(2):DIM B$(2):DIM C$(2):DIM D
$(2):DIM A(8,8):DIM S(2):DIM P$(2)
1050 A$(1)=GR$(0)+GR$(0):A$(2)=GR$(0)+GR
$(0)
1060 B$(1)=GR$(1)+GR$(2):B$(2)=GR$(3)+GR
$(4)
1070 C$(1)=GR$(5)+GR$(6):C$(2)=GR$(7)+GR
$(8)
1080 D$(1)=GR$(9)+GR$(10):D$(2)=GR$(11)+
GR$(12)
1090 S(1)=0:S(2)=0:START=1
1100 RETURN
1110 REM *** CARATTERI GRAFICI ***
1120 DEFGR$(0)=255,129,129,129,129,129,1
29,255
1130 DEFGR$(1)=255,129,129,129,131,135,1
58,252
1140 DEFGR$(2)=255,129,129,129,1,1,1,15
1150 DEFGR$(3)=248,128,128,128,129,129,1
29,255
1160 DEFGR$(4)=63,113,225,193,129,129,12
9,255
1170 DEFGR$(5)=255,129,129,129,129,128,1
28,240
1180 DEFGR$(6)=255,129,129,129,193,225,1
13,63
1190 DEFGR$(7)=252,142,135,131,129,129,1
29,255
1200 DEFGR$(8)=15,1,1,1,129,129,129,255
1210 DEFGR$(9)=255,128,128,128,128,128,1
28,128
1220 DEFGR$(10)=255,1,1,1,1,1,1,1
1230 DEFGR$(11)=128,128,128,128,128,128,
128,255
1240 DEFGR$(12)=1,1,1,1,1,1,1,255
1250 RETURN

```




AGENDA APPUNTAMENTI



Pubblichiamo in questo numero l'agenda appuntamenti, quarta ed ultima parte del pacchetto di programmi per la gestione domestica: con il prossimo numero della rivista saranno presentati i listati dei tre programmi che lo completano nonché le necessarie istruzioni e modifiche da apportare a quelli già pubblicati e che ne consentono il lancio automatico e la gestione mediante un menu generale.

L'agenda appuntamenti gestisce le annotazioni per un intero anno, separate giorno per giorno; le stringhe sono collocate su un massimo di 250 righe di 76 caratteri ciascuna, per un totale di quasi 20.000 caratteri disponibili al giorno, largamente sufficienti non solo per registrare gli appuntamenti, ma anche per tenere un diario molto dettagliato.

La ricerca dei dati è molto veloce e prevede solo l'inserimento di giorno e mese richiesti; sotto forma numerica i dati sono infatti archiviati in 12 subdirectories ciascuna intestata ad un mese dell'anno. La ricerca può essere automaticamente continuata fino a selezionare il contenuto del primo giorno utilizzato, successivo a quello visualizzato. Ogni schermata presenta fino a 25 righe di testo; i tasti del cursore consentono di sfogliare l'agenda sia in avanti che indietro.

L'intera agenda o parte di essa può essere stampata su carta.

Al termine dell'anno corrente, l'opzione «PREDISPOSIZIONE NUOVO ANNO» annulla tutte le registrazioni archiviate predisponendo i files per nuove immissioni per conservare le annotazioni è quindi necessario stamparle su carta.

Istruzioni per l'uso del Programma

Caricate il programma selezionando le directory principali (*DIR\$) e inserendo il comando LOAD"AA". Quando il programma viene usato per la prima volta, occorre procedere «una tantum» alla sua inizializzazione inserendo l'anno corrente: vengono automaticamente create le subdirectories per memorizzare i dati di ogni mese (DIR\$.AGEAPP.GEN,...), mentre l'anno è inserito nel file denominato "00" e viene letto ad ogni avvio di esecuzione.

Consultazione

Per consultare l'agenda occorre inserire la data richiesta (giorno e mese in forma numerica): sono visualizzate le prime 25 righe di testo archiviate e le successive, ove presenti, possono essere stampate usando i comandi del cursore; un breve suono segnala la fine del testo. Se, in corrispondenza della data immessa, non è presente alcuna annotazione, il programma ne segnala l'assenza. In entrambi i casi, premendo il tasto 'COPY' viene attivata la ricerca automatica in archivio delle annotazioni immediatamente successive in ordine cronologico: la ricerca può essere arrestata in qualunque momento del suo svolgimento premendo il tasto "I".

Aggiornamento

Dopo la scelta della data appare il testo eventualmente già archiviato, al quale si possono aggiungere altre righe fino ad un massimo di 250. L'interruzione dell'immissione avviene inserendo il carattere "I" all'inizio di una nuova riga.

Correzione/Cancellazione

Il testo memorizzato può essere corretto o cancellato in qualunque momento: dopo aver inserito la data, i dati archiviati sono visualizzati, riga per riga a partire dalla prima. Il tasto 'COPY' conferma il testo passando alla riga successiva, 'DELETE' provoca la cancellazione dell'intera riga, mentre la pressione di 'C' ne permette la correzione: il contenuto viene stampato in reverse e su di esso si può sovrascrivere l'intera riga corretta. Il tasto "I" permette di uscire in qualunque momento dalla sequenza ed i dati corretti passano in registrazione che, comunque, procede automaticamente dopo la visualizzazione dell'ultima riga memorizzata. Questa opzione, pertanto, non permette l'aggiunta di ulteriori righe di testo, operazione gestita dall'opzione precedente.

Stampa

L'opzione prevede l'inserimento di una data iniziale e finale: il programma ricerca automaticamente ogni data cui corrisponde almeno una riga di testo e limita ad esse la stampa. Usando il modulo continuo il passaggio alla pagina successiva avviene automaticamente dopo 50 righe di testo; la stessa procedura viene seguita con il foglio singolo, con la differenza che, dopo l'inserimento di una nuova pagina, occorre premere il pulsante 'LOCAL' sulla stampante. La stampa può essere arrestata tenendo premuto il tasto "I" fino ad avvenuta interruzione.

Predisposizione Nuovo Anno

Poiché la cronologia delle registrazioni non tiene conto dell'anno, ma solamente del mese e del giorno, è necessario, al termine dell'anno stesso, procedere ad una reinizializzazione dell'archivio che eviti il richiamo di annotazioni dell'anno precedente presentate come relative al periodo corrente. Tale opzione (la cui esecuzione, per il suo carattere distruttivo, deve essere prima confermata) annulla tutte le annotazioni memorizzate ed incrementa automaticamente l'anno corrente.

STRUTTURA DEL PROGRAMMA

R.60-100	Comandi preliminari e dimensionamento vettori.
R.110.	Lettura anno corrente o inizializzazione.
R.120-240	Visualizzazione menu e scelta dell'opzione.
R.250-400	Procedura di inizializzazione del programma.
R.410-550	Procedura di inserimento della data con controllo automatico degli anni bisestili e dei mesi di 30 giorni.
R.560-700	Procedura di scelta della directory relativa al mese inserito.
R.710-800	Procedura di stampa su carta delle righe di testo.
810-830	Procedura per la stampa di un messaggio un carattere alla volta.
R.840-860	Procedura per la costruzione della stringa che visualizza la data.
R.870-960	Procedura di controllo in caso di incremento automatico della data.
R.970-1220	Opzione di consultazione.
R.1230-1330	Opzione di aggiornamento.
R.1340-1540	Opzione di correzione/cancellazione.
R.1550-1710	Opzione di stampa su carta.
R.1720-1870	Opzione per la predisposizione nuovo anno.
R.1880-1900	Opzione di uscita dal programma.
R.1910	Rinvio dell'esecuzione al menu in caso di errori.

Uscita

Consente di uscire dal programma al termine della sua utilizzazione: è consigliabile eseguire sempre questa opzione prima di spegnere il computer. Come già detto nella presentazione del programma di contabilità domestica (n. 5 di maggio 1987 di LIST), quando il pacchetto sarà completato con il lancio automatico ed il menu generale, l'opzione uscita rinvierà automaticamente a quest'ultimo.

AVVERTENZE IMPORTANTI

Prima di iniziare la trascrizione del listato, leggete attentamente i paragrafi che seguono:

1 - I dati sono archiviati sul dischetto in una subdirectory che è necessario creare da tastiera prima di far girare il programma che, altrimenti, si bloccherebbe non trovando lo spazio su cui memorizzare i dati. Pertanto, la prima operazione da eseguire è l'inserimento dei seguenti comandi:

> DIR\$
> CDIR\$.AGEAPP

2 - Poiché l'agenda appuntamenti è inserita in un più ampio pacchetto di programmi gestibili mediante un menu principale, è necessario, affinché possa essere caricato automaticamente, assegnargli un nome preciso. Pertanto la registrazione del programma su disco va eseguita nella directory principale assegnando al filo il nome "AA" i comandi necessari sono i seguenti:

> *DIR\$
> SAVE "AA"

3 - Il comando *FX4,1 di R.60 modifica i codici prodotti dai comandi del cursore e dal tasto 'COPY' i quali, non appena il programma è passato in esecuzione, perdono la propria funzione originale per l'edit del testo. Se si rende quindi necessario modificare il contenuto di una riga di programma, occorre, per evitare di riscriverla tutta, inserire il comando *FX4,0 che ripristina le normali funzioni dei tasti cursore e copy. Il comando può essere naturalmente inserito in uno dei tasti funzione (ad esempio f0) in modo da eseguirlo semplicemente premendo il tasto funzione stesso:

> *KEY0*FX4,0:M

4 - Il comando *FX200,3 di R.70 provoca la disattivazione del tasto ESCAPE e la cancellazione della memoria con il tasto BREAK: tale precauzione è utile per evitare che una accidentale pressione del tasto ESCAPE provochi una uscita del programma con conseguente perdita dei dati inseriti e non ancora registrati. Poiché in tale configurazione non è possibile uscire dal programma per correggere righe contenenti errori di copiatura, vi consigliamo di inserire la R.70 solo dopo aver trascritto e collaudato tutto il programma, avendo la certezza della completa assenza di errori.



LIST

1

```
>L.
10REM***          ***
20REM***AGENDA APPUNTAMENTI***
30REM*** EN.FA.SOFT-1987 ***
40REM***      PC-128  S      ***
50REM***          ***
60*FX4,1
70*FX200,3
80ON ERROR GOTO 1910
90DIM A$(251)
100*DIR$.AGEAPP
110A=OPENIN"00":IF A=0 THEN MODE
128:PROCiniz ELSE INPUT#A,AN$:CLOS
E#A
120DATA Consultazione,Aggiorname
nto,Correzione/cancellazione,Stamp
a,Predisposizione nuovo anno,Uscit
a
130RESTORE 120:FOR K=1 TO 6:READ
A$(K):NEXT
140MODE 135:PRINT TAB(8)CHR$141;
CHR$129;"AGENDA APPUNTAMENTI":PRIN
T TAB(8)CHR$141;CHR$129;"AGENDA AP
PUNTAMENTI"
150FOR K=1 TO 6:PRINT TAB(0,K+4)
CHR$156;CHR$135;" ";A$(K):NEXT:P=
1:PP=1
160PRINT TAB(0,22)CHR$133;CHR$15
7;CHR$131;CHR$136;"Selezionare coi
comandi del cursore":PRINT CHR$13
3;CHR$157;CHR$131;CHR$136;"e preme
re 'RETURN'"
170PRINT TAB(0,PP+4)CHR$156;CHR$
137;" ";A$(PP):PRINT TAB(0,P+4)CH
R$132;CHR$157;CHR$130;CHR$136;A$(P
)
180A=GET
190IF A=138 THEN 230
200IF A=139 THEN 240
210IF A=13 THEN ON P GOTO 970,12
30,1340,1550,1720,1880
220GOTO 180
230IF P<6 THEN PP=P:P=P+1:GOTO 1
70 ELSE P=6:GOTO 170
240IF P>1 THEN PP=P:P=P-1:GOTO 1
70 ELSE P=1:GOTO 170
250DEFPROCiniz
260PRINT TAB(32)"INIZIALIZZAZION
E":INPUT""Inserire l'anno corren
te: ";AN$:IF LEN(AN$)<>4 THEN SOUN
D1,-5,20,10:CLS:GOTO 260
270A=OPENOUT"00":PRINT#A,AN$:CLO
SE#A
280*CDIR$.AGEAPP.GEN
290*CDIR$.AGEAPP.FEB
300*CDIR$.AGEAPP.MAR
310*CDIR$.AGEAPP.APR
320*CDIR$.AGEAPP.MAG
330*CDIR$.AGEAPP.GIU
340*CDIR$.AGEAPP.LUG
350*CDIR$.AGEAPP.AGO
360*CDIR$.AGEAPP.SET
```

LIST

2

```
370*CDIR$.AGEAPP.OTT
380*CDIR$.AGEAPP.NOV
390*CDIR$.AGEAPP.DIC
400ENDPROC
410DEFPROCins_data
420PRINT""
430INPUT"Giorno (/=USCITA): ";G:
IF G=0 THEN 550
440IF G<0 OR G>31 THEN SOUND1,-5
,20,10:VDU11:PRINT TAB(19)" ":
VDU11:GOTO 430
450INPUT"Mese: ";M:
IF M<1 OR M>12 THEN SOUND1,-5,20,1
0:VDU11:PRINT TAB(19)" ":VDU11
:GOTO 450
460IF G<29 THEN 540
470IF G=29 AND M=2 THEN 510
480IF G>29 AND M=2 THEN 530
490IF G=31 THEN 520
500GOTO 540
510IF INT(VAL(AN$)/4)<>VAL(AN$)/
4 THEN 530 ELSE 540
520IF M=4 OR M=6 OR M=9 OR M=11
THEN 530 ELSE 540
530SOUND1,-5,20,10:VDU11:VDU11:P
RINT" ":PR
INT" ":VDU
11:VDU11:GOTO 430
540PROCdata
550ENDPROC
560DEFPROCsc_dir
570ON M GOTO 580,590,600,610,620
,630,640,650,660,670,680,690
580SCLI"DIR$.AGEAPP.GEN":GOTO 7
00
590SCLI"DIR$.AGEAPP.FEB":GOTO 7
00
600SCLI"DIR$.AGEAPP.MAR":GOTO 7
00
610SCLI"DIR$.AGEAPP.APR":GOTO 7
00
620SCLI"DIR$.AGEAPP.MAG":GOTO 7
00
630SCLI"DIR$.AGEAPP.GIU":GOTO 7
00
640SCLI"DIR$.AGEAPP.LUG":GOTO 7
00
650SCLI"DIR$.AGEAPP.AGO":GOTO 7
00
660SCLI"DIR$.AGEAPP.SET":GOTO 7
00
670SCLI"DIR$.AGEAPP.OTT":GOTO 7
00
680SCLI"DIR$.AGEAPP.NOV":GOTO 7
00
690*DIR$.AGEAPP.DIC
700ENDPROC
710DEFPROCstmp
720F$=STR$(X):PRINT"
":VDU11:PRINT"Data: ";F$;"/";M
;"/";AN$:VDU11
730A=OPENIN(F$):IF A=0 THEN 800
```


LIST

3

```

740INPUT#A,J:IF J=0 THEN CLOSE#A
:GOTO 800
750VDU2:VDU21
760FOR K=1 TO J:INPUT#A,A$(K):NE
XT:CLOSE#A
770PRINT CHR$1;CHR$14;F$;"/";M;
"/";AN$:CL=CL+2:PRINT:FOR K=1 TO J:
PRINT A$(K):CL=CL+1:IF CL>50 THEN
CL=0:PRINT CHR$1;CHR$12
780NEXT K:PRINT:PRINT:CL=CL+2:IF
CL>50 THEN CL=0:PRINT CHR$1;CHR$1
2
790VDU6:VDU3
800ENDPROC
810DEFFPROCcar
820FOR MS=1 TO LEN(T$):PRINT MID
$(T$,MS,1)::Z=INKEY(10):NEXT MS
830ENDPROC
840DEFFPROCdata
850F$=STR$(G):DT$=STR$(G)+"/"+ST
R$(M)+"/"+AN$+" "
860 ENDPROC
870DEFFPROCctrl_data
880IF G<29 THEN 960
890IF G=29 AND M=2 THEN 930
900IF G>29 AND M=2 THEN 950
910IF G=31 THEN 940
920GOTO 960
930IF INT(VAL(AN$)/4)<>VAL(AN$)/
4 THEN 950 ELSE 960
940IF M=4 OR M=6 OR M=9 OR M=11
THEN 950 ELSE 960
950G=32
960ENDPROC
970REM***CONSULTAZIONE***
980MODE 128:Z=0:J=0:VDU19,1,2,0,
0,0:PRINT TAB(33)"CONSULTAZIONE":P
ROCins_data:IF G=0 THEN 130
990PROCsc_dir
1000VDU26:PRINT TAB(10)"CONSULTAZ
IONE";TAB(65)DT$:PRINT TAB(0,30)CH
R$138;"=Avanti ";CHR$139;"=Indi
etro 'COPY'=Date successive
/=Uscita"
1010VDU28,0,29,79,1:CLS:A=OPENIN(
F$):IF A=0 THEN 1150
1020INPUT#A,J:IF J=0 THEN CLOSE#A
:GOTO 1150
1030SOUND1,-5,200,5:FOR K=1 TO J:
INPUT#A,A$(K):NEXT:CLOSE#A:K=1
1040CLS:PRINT:FOR R=K TO K+25:IF
R>J THEN 1060
1050PRINT A$(R)
1060NEXT
1070Z=GET:IF Z=47 THEN 980
1080IF Z=138 THEN K=K+20:GOTO 112
0
1090IF Z=139 THEN K=K-20:GOTO 113
0
1100IF Z=135 THEN 1160
1110GOTO 1070
1120IF K>J THEN K=J-25:SOUND1,-5,

```

LIST

4

```

H=47 THEN 980
1190GOTO 1000
1200M=M+1:IF M>12 THEN 1220
1210G=1:PROCsc_dir:GOTO 1180
1220PRINT""Nessun'altra registr
azione presente in archivio per l'
anno corrente.":SOUND1,-5,20,10:Z=
INKEY(200):GOTO 980
1230REM***AGGIORNAMENTO***
1240MODE 128:VDU19,1,3,0,0,0:PRIN
T TAB(33)"AGGIORNAMENTO":PROCins_d
ata:IF G=0 THEN 130
1250CLS:PRINT TAB(10)"AGGIORNAMEN
TO";TAB(65)DT$:PRINT TAB(0,30)"Per
interrompere l'immissione inserir
e una '/' a inizio riga."
1260VDU28,0,29,79,2:PROCsc_dir:A=
OPENIN(F$):IF A=0 THEN J=0:GOTO 13
00
1270INPUT#A,J:IF J=0 THEN CLOSE#A
:GOTO 1300
1280IF J=250 THEN PRINT""Capaci
ta' di memoria esaurita.":SOUND1,-
5,20,10:CLOSE#A:Z=INKEY(200):GOTO
130
1290PRINT:FOR K=1 TO J:INPUT#A,A$
(K):PRINTA$(K):NEXT:CLOSE#A
1300K=J:PRINT:REPEAT:K=K+1:IF K>2
50 THEN 1320
1310PRINT TAB(1)"
":VDU11:
INPUTLINE A$(K):IF LEN(A$(K))>76 T
HEN SOUND1,-5,20,10:VDU11:GOTO 131
0
1320UNTIL K>250 OR A$(K)="/"
1330J=K-1:A=OPENOUT(F$):PRINT#A,J
:FOR K=1 TO J:PRINT#A,A$(K):NEXT:C
LOSE#A:SOUND1,-5,200,5:GOTO 1240
1340REM***CORREZIONE/CANCELLAZION
E***
1350MODE 128:VDU19,1,6,0,0,0:PRIN
T TAB(28)"CORREZIONE/CANCELLAZIONE
":PROCins_data:IF G=0 THEN 130
1360CLS:PRINT TAB(10)"CORREZIONE/
CANCELLAZIONE";TAB(65)DT$:PRINT TA
B(0,30)"'COPY'=Prossima riga 'C'=
Correzione riga 'DELETE'=Cancella
zione riga /=Uscita";
1370VDU28,0,29,79,1:PROCsc_dir:A=
OPENIN(F$):IF A=0 THEN 1520
20,10
1130IF K<1 THEN K=1:SOUND1,-5,20,
10
1140GOTO 1040
1150IF Z<>135 THEN PRINT""Nessu
na registrazione presente in archi
vio per la data richiesta.":SOUND1
,-5,20,10:GOTO 1070
1160G=G+1:PROCctrl_data
1170IF G>31 THEN 1200
1180PROCdata:VDU26:H=INKEY(1):IF

```


LIST

5

```

1380INPUT#A,J:IF J=0 THEN CLOSE#A
:GOTO 1520
1390FOR K=1 TO J:INPUT#A,A$(K):NE
XT:CLOSE#A
1400K=0:PRINT:REPEAT:K=K+1:PRINT
A$(K)
1410Z=GET:IF Z=47 THEN 1460
1420IF Z=135 THEN 1460
1430IF Z=67 OR Z=99 THEN 1530
1440IF Z=127 THEN A$(K)="":VDU11:
PRINT"
":VDU11:GOTO 1460
1450GOTO 1410
1460UNTIL K=J OR Z=47
1470B=0:FOR K=1 TO J:IF A$(K)=""
THEN 1490
1480B=B+1
1490NEXT:A=OPENOUT(F$):PRINT#A,B:
FOR K=1 TO J:IF A$(K)="" THEN 1510
1500PRINT#A,A$(K)
1510NEXT:CLOSE#A:SOUND1,-5,200,5:
GOTO 1350
1520PRINT""Nessuna registrazione
e presente in archivio." :SOUND1,-5
,20,10:Z=INKEY(200):GOTO 1350
1530VDU11:PRINT TAB(1)"
":
VDU11:COLOUR0:COLOUR129:PRINT TAB(
1)A$(K):COLOUR1:COLOUR128:VDU11:IN
PUTLINE B$:IF LEN(B$)>76 THEN SOUN
D1,-5,20,10:GOTO 1530
1540A$(K)=B$:GOTO 1460
1550REM***STAMPA***
1560MODE128:VDU19,1,2,0,0,0:PRINT
TAB(37)"STAMPA"
1570VDU28,0,31,79,1:CLS:PRINT""D
ata iniziale stampa":PROCins_data
:IF G=0 THEN 130
1580G1=G:M1=M:PRINT""Data fina
le stampa":PROCins_data:IF G=0 TH
EN 130
1590G2=G:M2=M:IF M2<=M1 AND G2<G1
THEN SOUND1,-5,20,10:GOTO 1570
1600CLS:PRINT""Stampa dal ";G1
;"/";M1;"/";AN$;" al ";G2;"/";M2;
"/";AN$
1610PRINT""Per interrompere
la stampa tenere premuto il tasto
'/' fino ad avvenuta":PRINT"inter
ruzione." :PRINT"
1620VDU2:VDU21:PRINTCHR$1;CHR$14;
TAB(14)"AGENDA ";AN$:PRINT:PRINT:
CL=2:VDU6:VDU3
1630FOR R=M1 TO M2:M=R:PROCsc_dir
:IF M1=M2 THEN GI=G1:GF=G2:GOTO 16
70
1640IF R=M1 THEN GI=G1:GF=31:GOTO
1670
1650IF R<M2 THEN GI=1:GF=31:GOTO
1670

```

LIST

6

```

1660GI=1:GF=G2
1670FOR X=GI TO GF:Z=INKEY(10):IF
Z=47 THEN X=GF+1:R=M2+1:GOTO 1700
1680G=X:PROCctrl_data:IF G>31 THE
N 1700
1690PROCstmp
1700NEXT X
1710NEXT R:GOTO 1570
1720REM***PREDISPOSIZIONE NUOVO A
NNO***
1730MODE 135:PRINTCHR$129;TAB(7)"
PREDISPOSIZIONE NUOVO ANNO":PRINT:
PRINTCHR$133;CHR$157;CHR$131;CHR$1
36;TAB(12)"ATTENZIONE!!":PRINTCHR$
135:PRINT
1740T$="Questo e' un comando dist
ruttivo." :PROCcar:PRINT:PRINT:T$="
Tutte le registrazioni saranno can
cella-te per predisporre l'anno "+
STR$(VAL(AN$)+1)+".":PROCcar:PRINT
:PRINT
1750PRINTCHR$132;CHR$157;CHR$130;
CHR$136;"Volete eseguire l'operazi
one? (S/N)"
1760Z$=GET$:IF Z$="S" OR Z$="s" T
HEN 1790
1770IF Z$="N" OR Z$="n" THEN 130
1780GOTO 1760
1790FOR R=1 TO 12:M=R:PROCsc_dir:
FOR X=1 TO 31:F$=STR$(X):A=OPENIN(
F$):IF A=0 THEN 1830
1800INPUT#A,J:IF J=0 THEN CLOSE#A
:GOTO 1830
1810CLOSE#A:A=OPENOUT(F$):PRINT#A
,O:CLOSE#A
1820PRINT TAB(0,23)CHR$130;"Cance
llazione dati del ";X;"/";R;"/";AN
$;" "
1830NEXT X
1840NEXT R
1850*DIR$.AGEAPP
1860A=OPENOUT"00":PRINT#A,STR$(VA
L(AN$)+1):CLOSE#A
1870PRINT TAB(0,23)"Predisposizio
ne eseguita " :SOUND1,-5,2
00,5:Z=INKEY(200):GOTO 100
1880REM***USCITA***
1890*DIR$
1900END
1910GOTO 130

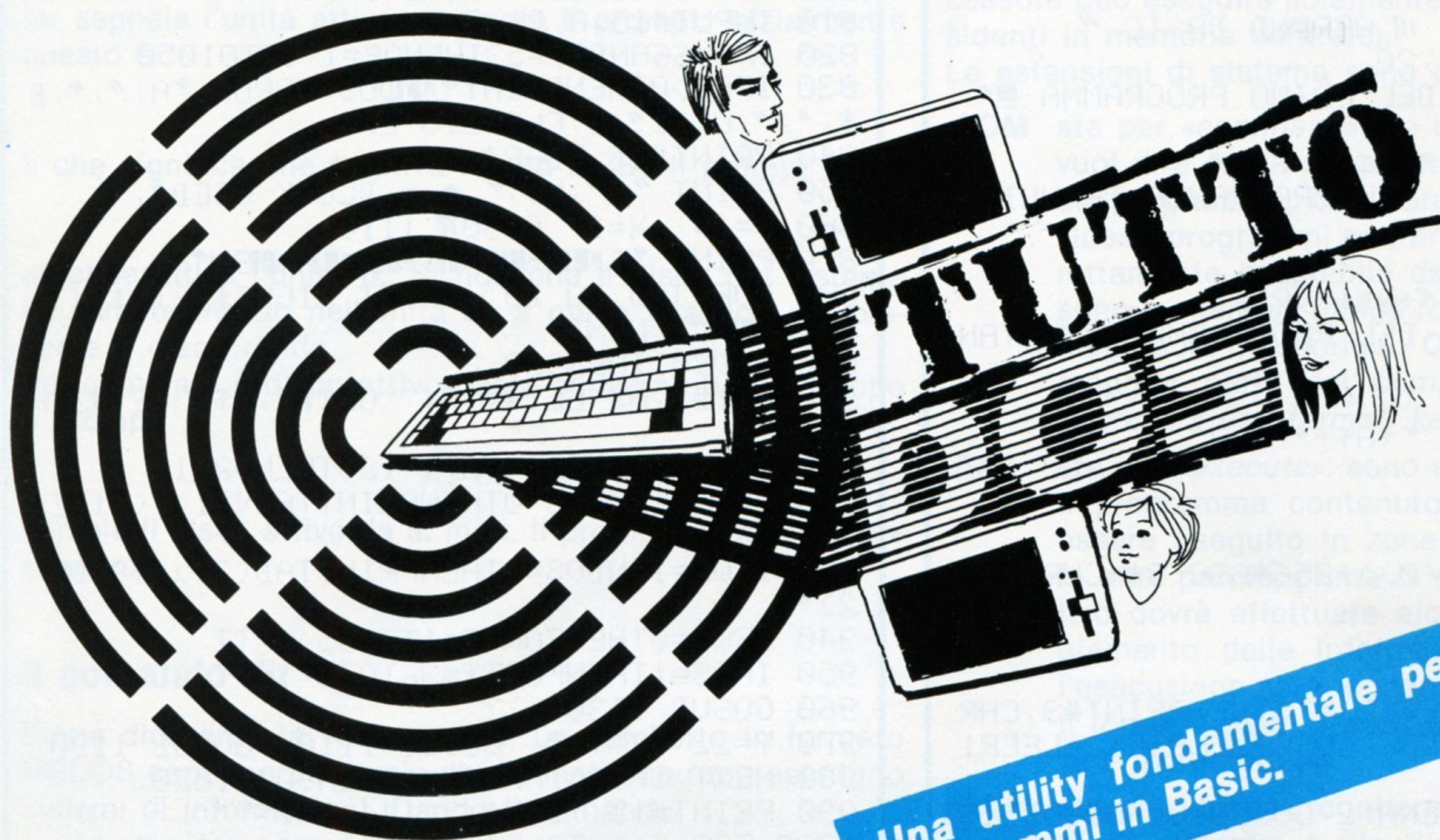
```



UN PACCHETTO DI PROGRAMMI
UNIFORMATO IN UNO

“DOMUS”

LO TROVERAI NEL PROSSIMO LIST



Una utility fondamentale per fondere due programmi in Basic.

Tutto DOS è un programma che implementa delle utilità per la gestione del Disk Drive e in più offre il MERGE o APPEND.

Dato il Run si avrà di fronte il menu principale comprendente 3 opzioni subito descritte:

OPZIONE 1: Si accede al sotto menu contenente le opzioni per la gestione del Disk Drive che sono:

- 1: Formattazione di un disco.
- 2: Cambia il nome della directory ad un disco.
- 3: Cancellazione di un file dal disco.
- 4: Invia un comando DOS diretto al Drive. I comandi possono essere l'inizializzazione del Drive "IO", il validate di un disco "VO".

Comunque per maggiori informazioni in merito consultare il Manuale del Drive.

- 5: Si ha la directory del disco corrente.
- 6: Si ritorna al menu principale.

OPZIONE 2: Con questa opzione si procede all'APPEND di due programmi BASIC.

NOTE: Il programma chiede in input il Nome del File sorgente, il Nome del File da unire ed il nome del File oggetto.

Attenzione: Il File da unire deve assolutamente avere i numeri di linea inferiori ai numeri di linea del File sorgente.

OPZIONE 3: Termine dell'elaborazione.

Questo programma gira unicamente sul C-128 ad 80 colonne e supporta come periferiche il Disk Drive. Notare che il presente programma è interamente gestito a finestre per far sì che l'utente possa in qualsiasi momento seguire al video tutte le operazioni da lui scelte.

LIST

1

```

10 REM *****
20 REM   MERGE PER C-128  80 COLONNE
30 REM DI ALESSANDRO CERACCHI SOFTWARE
40 REM   LIST  1987  LIST
50 REM *****
60 FAST
70 C$=""
  |"
80 C1$=""
  |"
90 C2$=""
  |"
100 C3$=""
  |"
110 SCLER:PRINT"***** 18 TUTTO DOS C-12
  8  80 COLONNE "
120 PRINTTAB(26)"DI ALESSANDRO CERACCHI
  1987"
130 WINDOW4,5,38,20
140 PRINT"
  |"
150 FORS=1TO13:PRINTC$:NEXTS
160 PRINT"
  |"
170 WINDOW5,6,37,18
180 PRINT"***** MAIN MENU" "
190 PRINT"  1 - OPERAZIONI DOS
200 PRINT"  2 - APPEND BASIC
210 PRINT"  3 - FINE LAVORO
220 PRINT"  4 SCELTA
230 GETKEYT$:IFT$="3"THENPRINT"*****":END
240 ONVAL(T$)GOSUB1750,250:GOTO180
250 WINDOW15,6,56,22
260 PRINT"
  |"

```



```

270 FORS=1TO14:PRINTC3$:NEXTS
280 PRINT"
"
290 PRINT"##### 10 APPEND BASIC "
300 WINDOW16,9,54,20
310 PRINT"##### NOME DEL PRIMO PROGRAMMA #####"
320 INPUTA$
330 OPEN1,8,2,"0:"+A$+",P,R"
340 PRINT"##### NOME DEL PROGRAMMA DA UNIRE
#####"
350 INPUTA$
360 OPEN2,8,3,"0:"+A$+",P,R"
370 PRINT"##### NOME DEL PROGRAMMA RISULTAN
TE #####"
380 INPUTA$
390 OPEN3,8,4,"00:"+A$+",P,W"
400 GET#1,A$,B$
410 GET#2,C$,D$
420 PRINT#3,A$B$;
430 S=ASC(A$+CHR$(0))+256*ASC(B$+CHR$(0))
440 A=1:GOSUB510
450 A=2:GOSUB510
460 IFL$(1)=" "ANDL$(2)=" "THENPRINT#3,CHR
$(0)CHR$(0);:CLOSE3:CLOSE2:CLOSE1:SLEEP1
:GOTO110
470 IFQ(2)<Q(1)THENA=2:GOSUB600:GOSUB510
:GOTO460
480 IFQ(1)<Q(2)THENA=1:GOSUB600
490 A=1:GOSUB510
500 GOTO460
510 L$(A)=" "
520 Q(A)=70000
530 GET#A,A$,B$,C$,D$
540 IFA$=" "ANDB$=" "ANDC$=" "THENRETURN
550 IFST>0THENRETURN
560 Q(A)=ASC(C$+CHR$(0))+256*ASC(D$+CHR$
(0))
570 GET#A,A$:IFA$=" "ORA$=CHR$(13)THENRET
URN
580 L$(A)=L$(A)+A$
590 GOTO570
600 S=S+LEN(L$(A)):W=INT(S/256):PRINT#3,
CHR$(S-256*W)CHR$(W);
610 PRINTS;Q(A)" ";
620 W=INT(Q(A)/256):PRINT#3,CHR$(Q(A)-25
6*W)CHR$(W)L$(A)CHR$(0);
630 RETURN
640 ::
650 Q2=0:Q3=0:Q4=0:F=0
660 PRINT"##### DISPLAY BLOCK ALLOCATI
ON MAP#####"
670 INPUT "##### DISK UNIT NR (8-15): ? 8 #####
#####";U
680 IF U<8 OR U>15 THEN PRINT "#####": GOT
O 670
690 INPUT "##### DISK DRIVE NR (0-1): ? 0#####
#####";D
700 IF D<0 OR D>1 THEN PRINT "#####": GOTO
690
710 PRINT "##### PUT TARGET DISK IN DRIVE";
D;"UNIT";U
720 PRINT " THEN PRESS ANY KEY TO START"
730 GET A$: IF A$=" " THEN 730
740 Z$=CHR$(0): D$=RIGHT$(STR$(D),1)
750 Y$="#####"
760 X$="#####
#####"
770 OPEN15,U,15,"U0>M1":CLOSE15
780 OPEN 15,U,15:PRINT#15,"I"+D$: OPEN

```

```

2,U,2,"#"
790 Q0=18:IFQ5=1THENQ0=53
800 PRINT#15,"U1:";2;D;Q0;0
810 INPUT#15,A,B$,C,D
820 IFA=66ANDQ0=53THENO2=1:GOTO1050
830 IFA<>0THENPRINT"#####DOS ERROR:"A";",",",B
$";",",",C";",",",D:CLOSE15:END
840 PRINT#15,"B-P";2;0
850 PRINT "#####Y$;" = _BLOCK USED";
860 Y=14: X=1: GOSUB 1110
870 PRINT "#####DISK ERROR#####"
880 FOR J=0 TO 20: PRINT RIGHT$(STR$(J),
2);"#####":NEXT
890 IFQ5=1THEN920
900 GET#2,A$:GET#2,A$: GET#2,A$: GET#2,A
$
910 IFASC(A$+CHR$(0))=128THENO2=1
920 IFQ2=1ANDQ5=0THENPRINTTAB(31)"(SIDE
1)";
930 IFQ2=1ANDQ5=1THENPRINTTAB(31)"(SIDE
2)";
940 IFQ5=0THENFORTT=1TO 35:T=TT
950 IFQ5=1THENFORTT=36TO70:T=TT-35
960 GOSUB 1130
970 Y=22: X=T+5: GOSUB 1110: GOSUB 1190
980 NEXT:IFQ2=1ANDQ5=1THENGOTO1020
990 PRINT#15,"B-P";2;144: N$=" "
1000 FOR J=1 TO 20: GET#2,A$: N$=N$+A$:
NEXT
1010 GOSUB 1280
1020 PRINT Y$;" PRESS ANY KEY ";
1030 GET A$: IF A$=" " THEN 1020
1040 IFQ2=1ANDQ5=0THENO2=1:GOTO790
1050 PRINT "#####DISK HEADER = ";CHR$(34)
;N$;CHR$(34)
1060 PRINT:PRINT;" BLOCKS FREE"
1070 IFQ2=1THENPRINT"#####1571 FORMAT DISK
.":PRINT" CAN ONLY READ SIDE ONE ON THIS
DRIVE."
1080 INPUT "##### VIEW ANOTHER DISKETTE (Y/
N) ? N#####";A$
1090 CLOSE 2: CLOSE 15: IFA$="Y"THEN RUN
1100 END
1110 PRINT LEFT$(Y$,Y);LEFT$(X$,X)"#####";
1120 RETURN
1130 IFQ5=1THEN1150
1140 GET#2,C$
1150 REM
1160 GET#2,A$: B(0)=ASC(A$+Z$)
1170 GET#2,A$: B(1)=ASC(A$+Z$)
1180 GET#2,A$: B(2)=ASC(A$+Z$): RETURN
1190 PRINT "#####": T$=RIGHT$(STR$(TT),2)
1200 PRINTLEFT$(T$,1);"#####";RIGHT$(T$,1);
"#####";
1210 L=20: IF T>17 THEN L=18
1220 IF T>24 THEN L=17
1230 IF T>30 THEN L=16
1240 FOR S=0 TO L
1250 Z=2+(S-INT(S/8)*8)AND(B(INT(S/8)))
1260 IF Z=0 THEN PRINT "#####": NEXT: RET
URN
1270 PRINT "+#####": NEXT: RETURN
1280 OPEN96,U,0,"$"+D$+"":
1290 FORXX=1TO34:GET#96,A$:NEXTXX
1300 GET#96,A$,B$:F=ASC(A$+CHR$(0))+256*
ASC(B$+CHR$(0))
1310 CLOSE96:RETURN
1320 ::
1330 PRINT"#####":WINDOW15,6,56,22
1340 PRINT"#####"

```


LIST

4

```

1350 FORS=1TO14:PRINTC3#:NEXTS
1360 PRINT"
"
1370 WINDOW16,7,54,20
1380 PRINT"  CAMBIA NOME AL DISCO "
1390 PRINT"  DISK UNITA' NR (8-15) ? 8
"
1400 INPUTU:IFUC8ORUD>15THENPRINT"TTT":GOTO1390
1410 PRINT"  DISK DRIVE NR (0/1) ? 0"
1420 INPUTD:IFD<0ORD>1THENPRINT"TTT":GOTO1410
1430 PRINT"  INSERISCI IL DISCO  U."U":D."D
1440 PRINT"  PREMI UN TASTO PER CONTINUARE"
1450 GETKEYT$
1460 A=1:V=16:Q$=CHR$(34):P$=CHR$(160)
1470 T=18:S=0:I$="I"+RIGHT$(STR$(D),1)
1480 OPEN1,U,15,I$:GOSUB1720
1490 GOSUB1680:F$=H$:GET#2,X$,X$,A$,B$
1500 PRINT"  NOME CORRENTE : ";Q$:F$:Q$
1510 PRINT"  ID : ";A$:B$
1520 PRINT"  NUOVO NOME (MAX 16 CAR.) 0
'QUIT'"
1530 INPUT" ? QUIT"QUIT":N$:L=LEN(N$):IFL>VTHEN1500
1540 IFN$="QUIT"THEN1740
1550 PRINT"  OK PER RISCRIVERE IL NOME ?
(S/N) ? S"
1560 INPUTX$:IFX$="S"THEN1500
1570 IFL=VTHEN1590
1580 FORX=L+1TOV:N$=N$+P$:NEXT
1590 PRINT#1,"B-P: ";2;144:PRINT#2,N$;
1600 PRINT#1,"U2: ";2;D:T:S:GOSUB1720
1610 PRINT#1,I$:GOSUB1720:CLOSE2
1620 PRINT"  VECCHIO NOME ";Q$:F$:Q$
1630 GOSUB1680:CLOSE2:CLOSE1
1640 PRINT"  NUOVO NOME ";Q$:H$:Q$
1650 PRINT"  VUOI CONTINUARE (S/N) ? S"
1660 INPUTX$:IFX$="S"THEN1430
1670 PRINT"  ":SLEEP2:GOTO1750
1680 OPEN2,U,2,"#":GOSUB1720:H$=" "
1690 PRINT#1,"U1: ";2;D:T:S:GOSUB1720
1700 PRINT#1,"B-P: ";2;144:FORX=ATOV
1710 GET#2,T$:H$=H$+T$:NEXT:RETURN
1720 INPUT#1,E,M$,J,K:IFE=0THENRETURN
1730 PRINT"  ERROR: ";E:M$:J,K
1740 CLOSE2:CLOSE1:SLEEP2:GOTO1750
1750 WINDOW26,8,58,24
1760 PRINT"
"
1770 FORS=1TO14:PRINTC1#:NEXTS
1780 PRINT"
"
1790 WINDOW27,9,56,22
1800 PRINT"  OPERAZIONI DOS "
1810 PRINT"  1 - FORMATTAZIONE DISCO
1820 PRINT"  2 - CAMBIA NOME DISCO
1830 PRINT"  3 - CANCELLA FILE
1840 PRINT"  4 - COMANDO DOS DIRETTO
1850 PRINT"  5 - DIRECTORY
1860 PRINT"  6 - RITORNO AL MENU
1870 GETKEYT$:IFT$="6"THEN110
1880 ONVAL(T$)GOTO1890,1320,2000,2100,2190

```

LIST

5

```

1890 WINDOW46,5,79,15
1900 PRINT"
"
1910 FORS=1TO8:PRINTC2#:NEXTS
1920 PRINT"
"
1930 WINDOW47,6,77,13
1940 PRINT"  FORMATTAZIONE DISCO "
1950 INPUT"  NOME DISCO: ";ND$:IFLEN(ND$)
=>0ORLEN(ND$)>16THEN1940
1960 PRINT"  OK ? ":GETKEYR$:IFR$="N"THEN1750
1970 IFR$="S"THENPRINT"  FORMATTAZIONE I
N CORSO..." :HEADER(ND$),IAC
1980 PRINT"  FINE FORMATTAZIONE..."
1990 SLEEP1:GOTO1750
2000 WINDOW46,5,79,15
2010 PRINT"
"
2020 FORS=1TO8:PRINTC2#:NEXTS
2030 PRINT"
"
2040 WINDOW47,6,77,13
2050 PRINT"  CANCELLAZIONE FILE "
2060 INPUT"  NOME FILE: ";NF$:IFLEN(NF$)
=>0ORLEN(NF$)>16THEN2050
2070 PRINT"  OK ? ":GETKEYR$:IFNF$="N"THEN1750
2080 IFR$="S"THENPRINT"  CANCELLAZIONE I
N CORSO..." :SCRATCH(NF$)
2090 GOTO1750
2100 WINDOW46,5,79,15
2110 PRINT"
"
2120 FORS=1TO9:PRINTC2#:NEXTS
2130 PRINT"
"
2140 WINDOW47,6,77,13
2150 PRINT"  COMANDO DOS DIRETTO "
2160 INPUT"  > ";CO$:IFCO$=" "THEN1750
2170 OPEN15,8,15,CO$:CLOSE15
2180 GOTO1750
2190 WINDOW46,5,79,15
2200 PRINT"
"
2210 FORS=1TO8:PRINTC2#:NEXTS
2220 PRINT"
"
2230 PRINT"  DIRECTORY DISCO "
2240 WINDOW47,8,77,13
2250 SLOW:CATALOG:FAST:SLEEP1:GOTO1750

```

ABBONATI A LIST

Farai un ottimo servizio di sostegno alla
TUA rivista



Un programma in linguaggio macchina per sdoppiare il vostro C-64, come se aveste sotto le mani due computers invece di uno.

Multitask 64 effettua un vero e proprio «sdoppiamento» della memoria del calcolatore, mettendo subito a disposizione due nuovi comandi Basic per gestire facilmente (sia da programma che in modo diretto) la nuova configurazione del C-64.

Prima di tutto si potrà disporre di due pagine video indipendenti, anziché di una sola. Il comando che gestisce le due pagine video ha la seguente sintassi:

`<-X,Y`

I parametri X ed Y possono assumere i valori 1 oppure 2: X rappresenta la pagina su cui si vuole scrivere, mentre Y rappresenta quale pagina si vuole visualizzare sullo schermo. Ad esempio: `<-2,2` permette di visualizzare la seconda pagina e di scriverci sopra. Ovviamente si può visualizzare una pagina e contemporaneamente scrivere sull'altra, usando, per esempio, il comando `<-2,1`: in questo modo ogni istruzione di PRINT che useremo produrrà una stampa sulla pagina due, ma verrà visualizzata la pagina uno. È inutile dire che questa possibilità può risultare utilissima in una infinità di applicazioni: in un programma con menu, ad esempio, invece di creare ogni volta la schermata con le varie opzioni, basterà stamparla una volta per tutte su una delle due pagine e poi richiamarla quando occorre, semplicemente visualizzando la pagina corrispondente.

Il secondo comando invece, permette di gestire due aree di memoria, ognuna contenente un differente programma. Si possono quindi caricare due programmi e conservarli contemporaneamente in memoria, scegliendo quale dei due far girare. La sintassi del comando corrispondente è la seguente:

`<-P1` oppure `<-P2`,

il cui significato è evidentemente quello di selezionare uno dei due programmi.

Come usare multitask 64

Tutti i comandi possono essere usati in modo diretto, cioè digitandoli sulla tastiera e facendoli eseguire subito con un return. Il comando relativo alla gestione delle pagine video può essere usato anche all'interno di un programma. In questo caso deve essere seguito, però, da una

istruzione di PRINT a vuoto, prima del PRINT vero e proprio. Una riga di programma potrebbe essere la seguente:

`10 <-1,2:PRINT:PRINT"MESSAGGIO"`

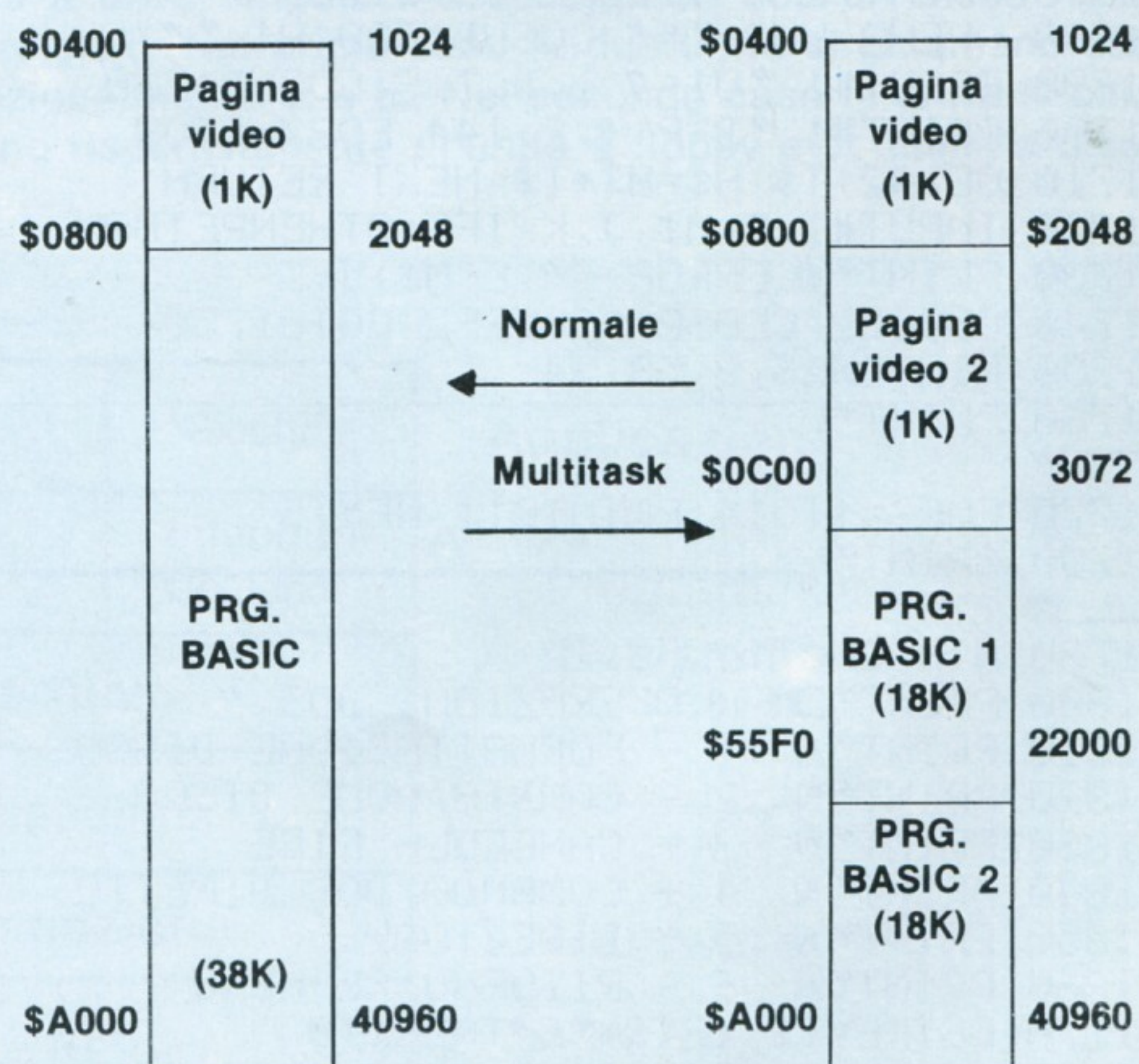
Per quanto riguarda il secondo comando, esso va usato evidentemente solo in modo diretto.

Prima di digitare o caricare da periferica un programma, bisogna fornire il comando `<-P1` o `<-P2` per selezionare l'area desiderata. Ed ora forniamo alcuni accorgimenti da seguire per utilizzare al meglio il MULTITASK 64:

1. Se il **turbo-tape** è resistente in memoria, appena verrà fatto girare il MULTITASK, i comandi non saranno più accettati.

Volendo far coesistere entrambi i sets di comandi, sarà sufficiente digitare una POKE 49244,104 dopo la SYS del MULTITASK;

Configurazione della memoria



2. Evitare di caricare programmi che occupano le locazioni dove risiede il codice macchina del MULTITASK (49152-49351), altrimenti quest'ultimo andrà perduto:

3. Usare le due aree P1 e P2 solo per memorizzare programmi in Basic, infatti essi sono rilocabili;

4. Se per qualche motivo (ad esempio, un resettaggio del C-64) venisse disattivato il MULTITASK, per riattivarlo occorrerà semplicemente digitare SYS 49152.

Per caricare in memoria il codice macchina vi forniamo un programma in Basic che effettua automaticamente anche il lancio della routine, se tutte le linee dei DATA sono state digitate correttamente.

Per i lettori che conoscono già il linguaggio macchina o per coloro che desiderano addentrarsi in questa affascinante tecnica di programmazione, pubblichiamo anche il listato in ASSEMBLER ed il relativo commento.

COMMENTO AL DISASSEMBLATO

C000 - C007: Viene spostato il puntatore dell'interprete Basic alla locazione d'inizio della nostra routine, ovvero C017.

C00A - C016: Inizializzazione delle 2 aree Basic.

C017 - C020: Lettura del primo byte del testo Basic: se non corrisponde al codice ASCII del simbolo <— (\$5F) allora il controllo viene restituito all'interprete Basic.

C023 - C038: Viene letto il carattere successivo del testo Basic e viene confrontato con i codici ACII dei simboli "1" oppure "2".

Viene quindi spostato il puntatore alla pagina video sulla quale il Basic lavora (\$0288) a seconda del numero letto.

C036 - C056: Viene letto un carattere dal testo e poi subito un altro, in modo da saltare la virgola presente nella sintassi del comando. L'ultimo byte letto viene al solito confrontato con i simboli 1 e 2 e il puntatore della pagina da visualizzare sullo schermo (\$D018) viene deviato sull'area opportuna.

C059 - C0A2: Test sul secondo byte letto per verificare che sia una 'P'. Se non lo è si salta alla locazione di syntax error (\$AF08), altrimenti si legge il carattere successivo per vedere se è un 1 o un 2.

A questo punto è necessario spostare tutti i puntatori in pagina zero che definiscono il testo Basic a seconda del programma scelto. Per far questo bisogna ogni volta conservare in un buffer i puntatori all'area prescelta. Questi due buffer sono stati allocati subito dopo la routine nelle locazioni \$C0AB - \$C0B5 e \$C0B6 - \$C0C3. Contengono i valori \$2B - \$3B.

LISTATO DISASSEMBLATO

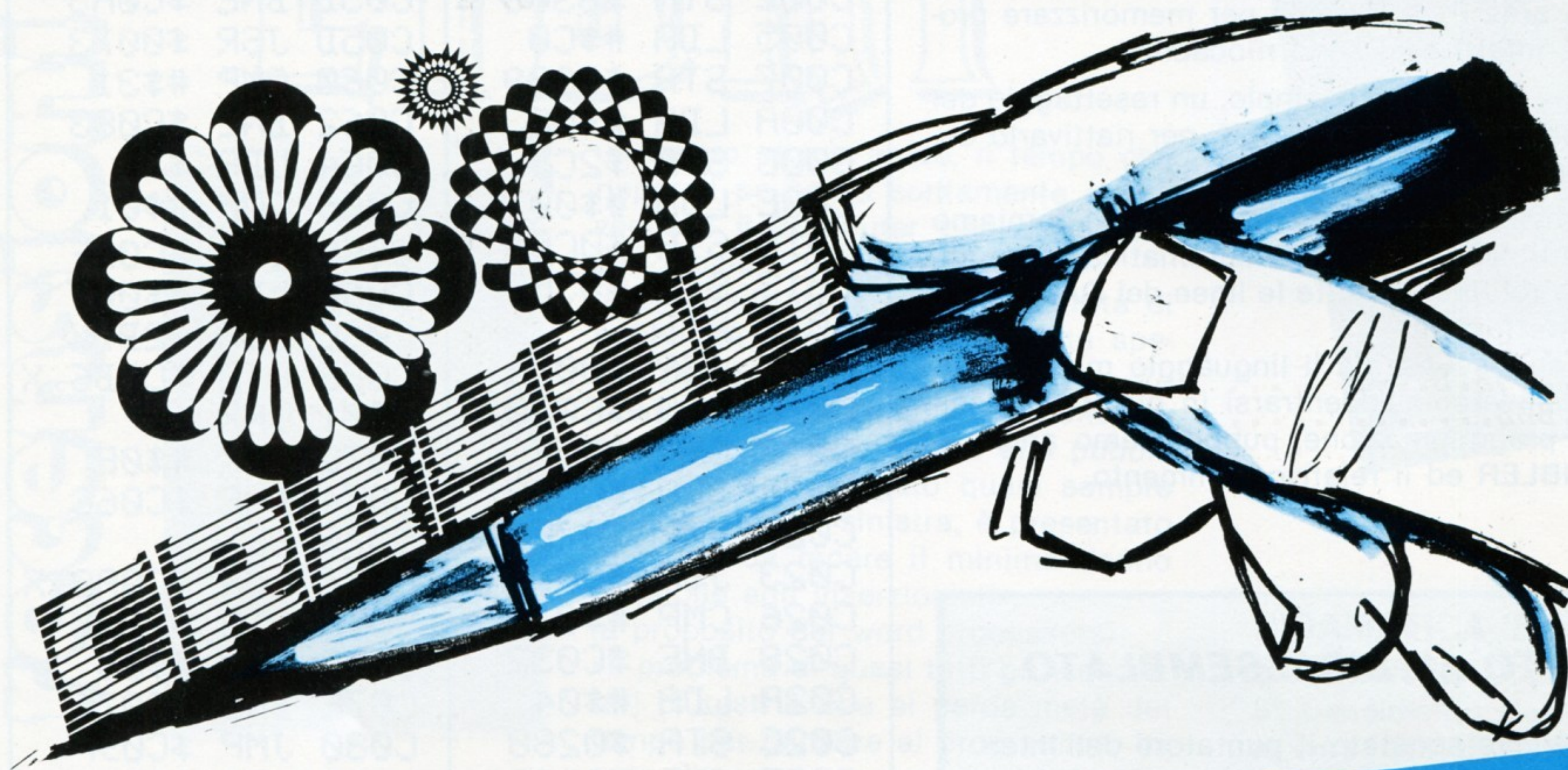
C000 LDA #\$17	C059 CMP #\$50
C002 STA \$0308	C05B BNE \$C0A5
C005 LDA #\$C0	C05D JSR \$0073
C007 STA \$0309	C060 CMP #\$31
C00A LDA #\$0C	C062 BNE \$C083
C00C STA \$2C	C064 LDA \$2B
C00E LDA #\$00	C066 CMP #\$01
C010 STA \$0C00	C068 BEQ \$C09F
C013 STA \$55F0	C06A LDX #\$00
C016 RTS	C06C LDA \$2B,X
C017 JSR \$0073	C06E STA \$C0B6,X
C01A BEQ \$C020	C071 INX
C01C CMP #\$5F	C072 CPX #\$0E
C01E BEQ \$C023	C074 BNE \$C06C
C020 JMP \$A7E7	C076 DEX
C023 JSR \$0073	C077 LDA \$C0A8,X
C026 CMP #\$31	C07A STA \$2B,X
C028 BNE \$C032	C07C CPX #\$00
C02A LDA #\$04	C07E BNE \$C076
C02C STA \$0288	C080 JMP \$C09F
C02F JMP \$C03B	C083 LDA \$2B
C032 CMP #\$32	C085 CMP #\$01
C034 BNE \$C059	C087 BNE \$C09F
C036 LDA #\$08	C089 LDX #\$00
C038 STA \$0288	C08B LDA \$2B,X
C03B JSR \$0073	C08D STA \$C0A8,X
C03E JSR \$0073	C090 INX
C041 CMP #\$31	C091 CPX #\$0E
C043 BNE \$C04D	C093 BNE \$C08B
C045 LDA #\$15	C095 DEX
C047 STA \$D018	C096 LDA \$C0B6,X
C04A JMP \$C09F	C099 STA \$2B,X
C04D CMP #\$32	C09B CPX #\$00
C04F BNE \$C0A5	C09D BNE \$C095
C051 LDA #\$25	C09F JSR \$0073
C053 STA \$D018	C0A2 JMP \$A7AE
C056 JMP \$C09F	C0A5 JMP \$AF08

LISTATO BASIC

```

10 FORI=49152TO49351:READA:POKEI,A:S=S+A
:NEXT:IFSC<>22600THENPRINT"ERRORE!" END
15 SYS49152:NEW
20 DATA 169,23,141,8,3,169,192,141,9,3,1
69,12,133,44,169,0,141,0,12,141,240,85
30 DATA 96,32,115,0,240,4,201,95,240,3,7
6,231,167,32,115,0,201,49,208,8,169,4
40 DATA 141,136,2,76,59,192,201,50,208,3
5,169,8,141,136,2,32,115,0,32,115,0,201
50 DATA 49,208,8,169,21,141,24,208,76,15
9,192,201,50,208,84,169,37,141,24,208
60 DATA 76,159,192,201,80,208,72,32,115,
0,201,49,208,31,165,43,201,1,240,53,162
70 DATA 0,181,43,157,182,192,232,224,14,
208,246,202,189,168,192,149,43,224,0
80 DATA 208,246,76,159,192,165,43,201,1,
208,22,162,0,181,43,157,168,192,232,224
90 DATA 14,208,246,202,189,182,192,149,4
3,224,0,208,246,32,115,0,76,174,167,76
100 DATA 8,175,1,12,3,12,3,12,3,12,240,8
5,0,0,240,85,241,85,243,85,243,85,243
110 DATA 85,0,160,0,0,0,160,26,76,106,19
5

```

Un autogeneratore di programmi grafici per il C 64 con l'utility Simon's basic.

Se il Simon's basic era già un valido aiuto per chi si accingeva alla realizzazione di programmi grafici, ora, con l'aggiunta del nostro programma le fatiche del programmatore sono veramente ridotte ai minimi termini! Infatti basterà disegnare con il joystick la schermata desiderata e sarà l'utility CRAJON a scrivere per voi il programma in Simon's basic corrispondente.

Il principio di funzionamento è simile a quello del programma autosprite, pubblicato sul numero di settembre: editare delle nuove linee di programma, mentre il programma stesso stà girando. Questa tecnica è chiamata tastiera dinamica, viene realizzata manipolando il buffer di tastiera.

Come si usa Crayon

Appena digitato il comando RUN, appare una schermata di presentazione corredata di alcune telegrafiche istruzioni per l'uso che ora approfondiremo insieme.

Premendo il tasto di return vengono richiesti i codici per il colore dello sfondo e del disegno, dopodiché apparirà la matita da muovere con il joystick per disegnare.

In basso a destra vengono visualizzate cinque lettere, ognuna delle quali rappresenta una specifica funzione che potrà essere attivata digitando il carattere corrispondente. In questo caso una piccola freccia indicherà che la funzione è diventata operativa. Esaminiamo le cinque funzioni:

'V' permette di spostarvi a vuoto con la matita, senza quindi disegnare e viene attivata automaticamente all'inizio del programma. Ovviamente basta premere il tasto **'V'** e la matita comincerà a disegnare.

'C' serve per tracciare dei cerchi: prima di tutto ci si posiziona con il joystick sul punto prescelto come centro della circonferenza e si preme il tasto **'C'**; in questo modo verrà disegnato un punto per fissare il centro. Poi bisogna allontanarsi dal centro a seconda del raggio desiderato e premere di nuovo la lettera **'C'**: verrà disegnata la circonferenza richiesta e il punto di centro sarà cancellato. È comodo attivare la funzione **'V'** durante l'esecuzione dei passi precedenti, in maniera tale che solo il cerchio sia tracciato.

'S' sceglie il colore del disegno se desideriamo cambiarlo durante l'esecuzione della schermata. La funzione **'P'** (PAINT) riempie di colore aree chiuse: basta posizionarsi con il joystick all'interno dell'area da riempire e premere il tasto **'P'**.

'L' traccia le linee: ci si posiziona sul punto di partenza e si preme **'L'**, poi ci si sposta sul punto di arrivo e si preme di nuovo **'L'**. Verrà così disegnata la retta passante per i due punti.

Per comunicare che il disegno è terminato, basta premere il pulsante di sparo del joystick e le nuove linee di programma saranno editate a partire dalla linea 1000. In pratica il programma edita queste linee anche durante tutta l'esecuzione del programma, ogni volta che il buffer di tastiera è pieno. Quindi non spaventatevi se periodicamente, mentre state disegnando, la schermata scompare e vengono visualizzate le linee di programma: in pochi secondi tutto torna a funzionare normalmente. Quando il disegno sarà ultimato, il programma risultante sarà contenuto a partire dalla linea 1000: tutte le linee precedenti potranno essere eliminate.

Ricordate, prima di digitare il listato, di caricare e lanciare il 'simon's basic', necessario al funzionamento di 'CRAYON'.

LIST

1

```

10 DATA 990
20 DESIGN 0,2176+49152
30 .....BB...
40 .....BBB...
50 .....BBB...
60 .....BBB...
70 .....BBB...
80 .....B...
90 .....B...B...
100 .....BBB.B...
110 .....BBB.B...
120 .....BBB.B...
130 .....BBB.B...
140 .....BBB...
150 .....BBB...
160 .....BBB...
170 .....BBB...
180 .....BBB...
190 .....BBB...
200 .....BBB...
210 .....BB...
220 .....B...

230 B.....
240 PRINT "CENTRE "CRAYON":PRINT:P
RINT:CENTRE "DI GIAN PIERO ROSI"
250 PRINT "PRINT"V= SPOSTAMENTO A VU
OTO":PRINT"OC= CERCHI"
260 PRINT"OS= SCELTA NUOVO COLORE":PRINT
"OP= PAINT (COLORA UN'AREA)"
270 PRINT"OL= LINEE":PAUSE "PREMI RE
TURN PER CONTINUARE...",500
280 COLOUR 2,2:POKE198,0:INPUT"VUOI
DISEGNARE (S/N)":R$:IFR$="N"THENEND
290 INPUT"COLORE DISEGNO E SFONDO":D
S,SF:READLI:Q=2
300 LI=LI+10:N=LI:GOSUB330:L$(0)=N$+" "
310 IFLI=1000THENN=SF:GOSUB330:L$(0)=L$(
0)+"HIRES"+STR$(DS)+"", "+N$
320 HIRES DS,SF:GOTO890
330 N$=STR$(N):N$=RIGHT$(N$,LEN(N$)-1):R
ETURN
340 X(1)=160:X(2)=100:X(3)=X(1):X(4)=X(2
):L=0:Q=2
350 TEXT 240,5,"SUPER DRAWER",1,1,6:
TEXT 240,170,"V C S P L",1,1,6
360 A$="V":GOSUB750:MOB SET 1,34,DS,0,0:
RLOCMOB 1,184,129,0,60
370 :
380 REM **** LETTURA JOY & KEY ****
390 :
400 GETA$:IFA$<" "THENGOSUB550:GOSUB750
410 B=JOY:IFB=0THEN400
420 IFB<>ATHENGOSUB550
430 IFB=10RB=20RB=8THENX(4)=X(4)-1
440 IFB>3ANDB<7THENX(4)=X(4)+1
450 IFB>5ANDB<9THENX(3)=X(3)-1
460 IFB>1ANDB<5THENX(3)=X(3)+1
470 IFB=128THEN700
480 IFVTHENX(1)=X(3):X(2)=X(4):A=0:GOTO5
00

```

LIST

2

```

490 PLOT X(3),X(4),1
500 RLOCMOB 1,X(3)+24,X(4)+29,0,1:GOTO40
0
510 :
520 :
530 REM **** CREAZIONE ISTRUZIONI ****
540 :
550 IFA=0THENA=B:RETURN
560 T$="":ONQGOTO570,590,600,610
570 T$="CIRCLE ":X(1)=XC:X(2)=YC:FORI=1T
O2:N=X(I):GOSUB330:T$=T$+N$+", "
580 NEXT N=R%:GOSUB330:T$=T$+N$+", "+N$+"
",":GOTO620
590 T$="LINE ":FORI=1TO4:N=X(I):GOSUB330
:T$=T$+N$+", ":NEXT:GOTO620
600 N=SF:GOSUB330:T$="LOW COL"+STR$(DS)+
", "+N$+", ":LOW COL DS,SF,1:GOTO620
610 N=X(4):GOSUB330:T$="PAINT"+STR$(X(3)
)+", "+N$+", ":GOTO620
620 Q=2:T$=T$+"1":X(1)=X(3):X(2)=X(4)
630 IFLN(L$(L))+LEN(T$)<75THEN650
640 L=L+1:LI=LI+10:N=LI:GOSUB330:L$(L)=N
$+" ":GOTO660
650 IFLN(L$(L))>5THENL$(L)=L$(L)+" "
660 L$(L)=L$(L)+T$:A=B:IFL<6THENRETURN
670 :
680 REM **** AUTO-EDITING LINEE ****
690 :
700 PRINT"FORI=0TOL:PRINTL$(I):NEXT
:FORI=631TO640:POKEI,13:NEXT:POKE198,10
710 PRINT"10 DATA"LI:PRINT"RUN":MOB OF
F 1:END
720 :
730 REM **** COMANDI TASTIERA ****
740 :
750 IFA$="V"THENV=NOTV:CHAR 240,180,30,A
BS(V),1:RETURN
760 IFA$="C"THENC=NOTC:CHAR 252,180,30,A
BS(C),1:ELSE:GOTO810
770 IFCTHENV1=V:V=-1:GOTO800
780 V=V1:RZ=SQR((XC-X(3))^2+(YC-X(4))^2)
:CIRCLE XC,YC,RZ,RZ,1:PLOT XC,YC,0:Q=1
790 GOTO870
800 XC=X(3):YC=X(4):PLOT XC,YC,1:RETURN
810 IFA$="S"THENNRM:MOB OFF 1:INPUT"COL
ORE DISEGNO":DS:ELSE:GOTO830
820 Q=3:CSET 2:MOB SET 1,34,DS,0,0:GOTO8
70
830 IFA$="P"THENQ=4:PAINT X(3),X(4),1:GO
TO870
840 IFA$="L"THENLN=NOTLN:CHAR 288,180,30
,ABS(LN),1:ELSE:RETURN
850 IFLNTHENV1=V:V=-1:GOTO800
860 V=V1:LINE XC,YC,X(3),X(4),1:X(1)=XC:
X(2)=YC
870 GOSUB560:RETURN
880 :
890 REM **** AUTO PROGRAMMA ****
900 :
3000 GOTO340

```

OGNI 15 DEL MESE... APPUNTAMENTO IN EDICOLA CON

LIST

PROGRAMMI PER IL TUO HOME COMPUTER



Come creare e ricreare figure a nostro piacere. Un vecchio passatempo sempre divertente.

Scopo di questo originalissimo videogioco è quello di costruire dei puzzles. Dopo aver dato il «RUN», apparirà una schermata di presentazione del gioco e successivamente verrà visualizzato un disegno che il C 64 scomporrà e toccherà a noi tentare di ricomporre. Naturalmente è possibile avere a nostra disposizione, accanto alla figura da ricomporre, la figura originale. Una volta terminato di comporre la prima figura ci verrà mostrato il tempo impiegato. Il gioco si snoderà così fino alla quarta figura. Finito di comporre la quarta ed ultima figura si avrà l'opportunità di creare delle nuove figure a nostro piacimento oppure di ripetere il gioco nel tentativo di migliorare il tempo di «ricomposizione» delle figure. Buon divertimento.

LIST

1

```

0 GOSUB378
2 PRINT"□":P=40:Q=P/2+2:R=0
4 IFW=0THENWW=1
6 IFW=1THENWW=2
8 IFW=2THENWW=3
10 IFW=3THENWW=4
12 IFFF=1THEN18
14 INPUT"VUOI LA FIGURA IN AIUTO ";W$
16 IFW$<>"S"ANDW$<>"N"THENRUN
18 PRINT"□ FIGURA "WW" 3PUZZLE■ "AD$
20 A$="XXXXXXXXXXXXXXXXXXXXXXXXXXXX":A=1147
22 PRINT"□ ABCDEFGH"
24 PRINT"      ───────────"
26 FORK=1TO8:PRINTK"13          ■1"K
28 NEXTK
30 PRINT"      ───────────"
32 PRINT"      ABCDEFGH"
34 IFFF=1THEN38
36 IFW$="S"THENGOSUB169
38 FORX=1TO8:FORY=1TO8
40 READA(X,Y):B(X,Y)=A(X,Y)
42 POKEA+P*Y+X,A(X,Y)
44 NEXT:NEXT
45 IFR=0THENR=1:PRINT"□□ PREPARO IL PUZZLE"
47 IFFF=1THEN344
48 IFW$="S"THENGOSUB180
50 DEFFN9(Z)=INT(RND(1)*8+1)

```

LIST

2

```

52 FORX=1TO8:FORY=1TO8
54 XX=FNA(Z):YY=FNA(Z)
56 IFB(XX,YY)=-1THEN54
58 C(X,Y)=B(XX,YY)
60 B(XX,YY)=-1:NEXT:NEXT
62 FORX=1TO8:FORY=1TO8
64 POKEA+P*Y+X,C(X,Y)
66 NEXT:NEXT
68 FORX=1TO8:FORY=1TO8
70 B(X,Y)=C(X,Y)
71 NEXT:NEXT
72 IFW$="S"THENPRINT"
"
73 IFW$="N"THENPRINT"
"
74 PRINTA$;

76 INPUT"RETURN PER COMINCIARE *";T$
78 TI$="000000"
80 PRINTA$;:FORU=1TO25:PRINT" ";:NEXT
82 PRINTA$;:INPUT"DA *****"X$
84 IFX$="*"THEN120
85 IFX$="HE"THENF1=1:GOSUB168:GOTO400
86 IFLEN(X$)>2THENPRINT"ITI":GOTO82
88 IFLEFT$(X$,1)<"A"ORLEFT$(X$,1)>"H"THE
N82
90 IFRIGHT$(X$,1)<"1"ORRIGHT$(X$,1)>"8"
HEN82
92 PRINTA$;
94 INPUT"A *****"
Y$
96 IFLEN(Y$)>2THEN82
98 IFLEFT$(Y$,1)<"A"ORLEFT$(Y$,1)>"H"THE
N92
100 IFRIGHT$(Y$,1)<"1"ORRIGHT$(Y$,1)>"8"
THEN92
102 X=ASC(LEFT$(X$,1))-64
104 Y=VAL(RIGHT$(X$,1))

106 X1=ASC(LEFT$(Y$,1))-64
108 Y1=VAL(RIGHT$(Y$,1))
110 POKEA+P*Y+X,C(X1,Y1)
112 POKEA+P*Y1+X1,C(X,Y)
114 C=C(X1,Y1):C1=C(X,Y)
116 C(X1,Y1)=C1:C(X,Y)=C
118 GOTO80

```


LIST

3

```

120 D=64
122 FORX=1TO8:FORY=1TO8
124 IFA(X,Y)<>C(X,Y)THEND=D-1
126 NEXT: NEXT
128 IFD=64THENT$=TI$
130 IFD=64THENPRINTA$;"HAI FINITO IN ";
132 IFD=64THENPRINTMID$(T$,3,2)" MINUTI
E ";
134 IFD=64THENPRINTMID$(T$,5,2)" SECONDI
"

136 IFD=64ANDW=0THENFORK=1TO3000:NEXT
138 IFD=64ANDW=0THENPRINT "J"
140 IFD=64ANDW=0THENW=1:GOTO2
142 IFD=64ANDW=1THENFORK=1TO3000:NEXT
144 IFD=64ANDW=1THENPRINT "J"
146 IFD=64ANDW=1THENW=2:GOTO2
148 IFD=64ANDW=2THENFORK=1TO3000:NEXT
150 IFD=64ANDW=2THENPRINT "J"
152 IFD=64ANDW=2THENW=3:GOTO2
154 IFD=64ANDW=3THENFORK=1TO3000:NEXT
156 IFD=64ANDW=3THENPRINT "J"
158 IFD=64ANDW=3THENW=4:GOTO2
160 IFD=64ANDW=4THENFORK=1TO3000:NEXT
162 IFD=64ANDW=4THENPRINT "FINE GIOCO"

164 IFD=64ANDW=4THEN252
166 GOTO80
168 IFW$="S"ANDX$="HE"THENF1=0:GOTO180
169 PRINT"800"TAB(0)"      ABCDEFGH"
171 PRINTTAB(0)"      "
172 FORK=1TO8
174 PRINTTAB(0)K"      "
176 PRINTTAB(0)"      "
178 PRINTTAB(0)"      ABCDEFGH"
180 FORX=1TO8:FORY=1TO8

182 POKEA+P*Y+X+0,A(X,Y)
184 NEXT: NEXT
186 RETURN
188 DATA 78,101,101,101,77,96,96,96
190 DATA 99,85,96,96,96,77,96,96
192 DATA 99,64,81,85,74,60,77,96
194 DATA 99,73,93,75,64,99,99,99
196 DATA 99,85,93,74,64,99,99,99
198 DATA 99,64,81,73,75,62,78,96
200 DATA 99,73,96,96,96,78,96,96
202 DATA 77,103,103,103,78,96,96,96
204 DATA 96,78,96,96,96,102,12,102
206 DATA 78,96,96,96,96,102,96,102
208 DATA 77,96,96,78,96,102,9,102
210 DATA 85,77,78,96,96,102,96,102
212 DATA 73,78,96,96,96,102,19,102
214 DATA 78,96,96,96,96,102,96,102
216 DATA 77,96,96,96,96,102,20,102
218 DATA 96,77,96,96,96,102,96,102
220 DATA 96,96,79,76,77,96,96,102
222 DATA 87,96,119,111,96,77,96,102
224 DATA 96,77,119,111,96,96,77,102

226 DATA 81,66,119,111,96,96,111,102
228 DATA 96,78,119,111,96,96,111,102
230 DATA 87,96,119,111,96,96,78,102
232 DATA 96,96,119,111,96,78,96,102

```

LIST

4

```

234 DATA 96,96,80,122,78,96,96,102
236 DATA 102,102,102,102,102,102,102,102
238 DATA 102,19,102,7,102,102,102,102
240 DATA 102,5,102,9,1,102,6,102
242 DATA 102,9,102,21,12,102,9,102
244 DATA 102,102,15,14,12,102,14,102
246 DATA 102,102,18,20,1,102,5,102
248 DATA 102,102,1,15,102,102,102,102
250 DATA 102,102,102,102,102,102,102,102

252 PRINT"      SOSTITUZIONE DEI VALOR
I DATA"
254 PRINT"    DOVRAI ORA DIGITARE I VALORI
NUMERICI    DEL CODICE POKE"
256 PRINT" PER POTER FORMARE UN'ALTRO DI
SEGNO "

258 PRINT" A PIACERE, DI TUO GRADIMENTO."

260 PRINT"-----"
262 PRINT"X-INSERISCI PRIMA IL NUMERO IN
CHR$"
264 PRINT" PREMERE LA VIRGOLA PER CONFER
MARE"
266 PRINT"X-SE VORRAI ANNULLARE LA LINEA
DOVRAI    PREMERE RETURN"
268 PRINT" INSERENDO NUOVAMENTE I VALORI
NUMERICI "
270 PRINT"X-F1 = NUOVA FIGURA"
272 PRINT"X-F3 = FINE "
274 GETR$:IFR$=""THEN274
276 IFR$="X"THEN354
278 IFR$="X"THENPRINT"J":END
280 GOTO274
282 PRINT"J":LN=AC:IN=2
284 POKE822,0:GOSUB316:POKE822,IN
286 LN=PEEK(820)+PEEK(821)*256:A$=STR$(L
N):D$=RIGHT$(A$,LEN(A$)-1)+ " D$ ":AS=0
288 A$="":PRINT"J"D$:IFAS=8THENAS=0:GOTO
302
290 GETK$:IFK$=""THEN290
292 IFK$=CHR$(20)ANDLEN(A$)>0THENA$=LEFT
$(A$,LEN(A$)-1):PRINTK$:GOTO290
294 IFK$=CHR$(13)THEN302
296 IFK$="," THEND$=D$+A$+K$:AS=AS+1:GOTO
318
298 IFK$<"0"ORK$>"9"THEN290
300 A$=A$+K$:PRINTK$:GOTO290
302 IFAS<>VAL(A$)THENPRINT"X      ** V
ALORI ANNULLATI **"
304 IFAS<>VAL(A$)THENPRINT"XRETURN = C
ANCELLI DALLA LINEA"LN"ALLA LINEA 192"
306 IFAS<>VAL(A$)THENPRINT"XDIVERSO DA
RETURN = INCREMENTO DALLA LINEA"LN;
308 IFAS<>VAL(A$)THENPRINT"ALLA LINEA 19
2"
310 IFAS<>VAL(A$)THENPOKE198,0:WAIT198,1
:RUN286
312 GOSUB316:PRINT"R /286":POKE631,19:POK
E632,13:POKE633,13:POKE634,13

314 POKE198,4:END

```


LIST

5

```

316 LN=LN+PEEK(822):H% =LN/256:POKE821,H%
:POKE820,LN-H%*256:RETURN
318 IFAS=8THEND$=LEFT$(D$,LEN(D$)-1)+CHR
$(13):GOTO288
320 IFLN=204THENFF=1:PRINT"FINE DATA "
:W=0:GOTO2
322 IFLN=220THENFF=1:PRINT"FINE DATA "
:W=0:GOTO2
324 IFLN=236THENFF=1:PRINT"FINE DATA "
:W=0:GOTO2
326 IFLN=252THENFF=1:PRINT"FINE DATA "
:W=0:GOTO2
328 GOTO288
330 PRINT"DEVI REGISTRARE IL NUOVO PROG
RAMMA"
332 PRINT"01) = SE HAI IL REGISTRATORE"

334 PRINT"02) = SE HAI IL DISK DRIVE"
336 GETG$:IFG$=""THEN336
338 IFG$="1"THENPRINT"1":SAVE"PUZZLE":RU
N

340 IFG$="2"THENPRINT"2"INOME DELLA FIGURA
":INPUTAD$:SAVE"PUZZLE "+AD$,8:RUN
342 GOTO336
344 PRINT"03) TUTTO OK"
346 GETAA$:IFAA$=""THEN346
348 IFAA$="S"THENFF=0:GOTO330
350 IFAA$="N"THENFF=0:GOTO252
352 GOTO346
354 PRINT"04) SCELTA FIGURA DA SOSTITUIRE"
356 PRINT"01) = PRIMA FIGURA"
358 PRINT"02) = SECONDA FIGURA"
360 PRINT"03) = TERZA FIGURA"
362 PRINT"04) = QUARTA FIGURA"
364 PRINT"05) QUALE"
366 GETRE$:IFRE$=""THEN366
368 IFRE$="1"THENAC=188:GOTO282
370 IFRE$="2"THENAC=204:GOTO282
372 IFRE$="3"THENAC=220:GOTO282
374 IFRE$="4"THENAC=236:GOTO282
376 GOTO366
378 REM *****

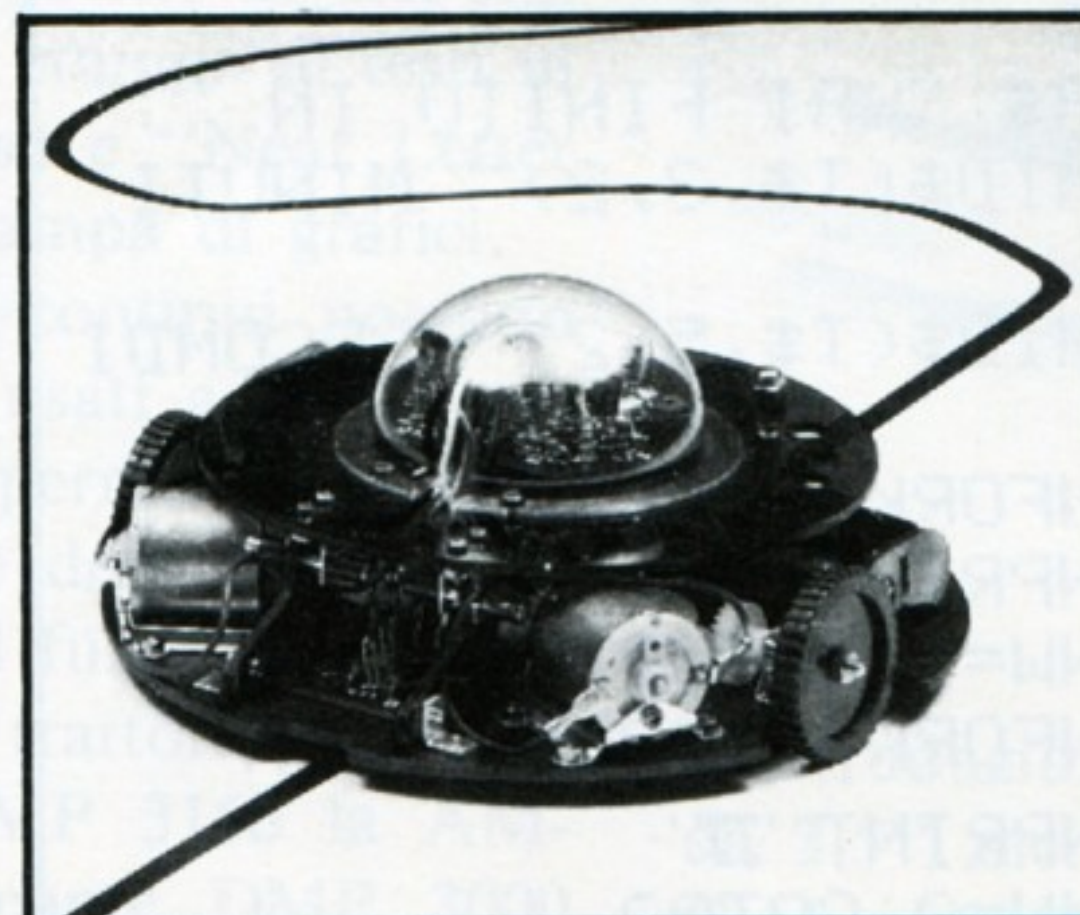
380 REM ***** I PUZZLE *****
382 REM ** DI **
384 REM ** ZERBINI MARCELLO & FABIO **
386 REM ** PER CBM 64 PER GLI AMICI **
388 REM ** DI LIST **
390 REM *****
392 PRINT"1"TAB(15)"I PUZZLE"
394 PRINTTAB(19)"I PUZZLE"
396 PRINTTAB(8)"ZERBINI MARCELLO & FABIO"

398 FORTT=1TO3000:NEXT:RETURN
400 IFF1=0THEN80
402 FORT=1TO5000:NEXT
404 FORX=1TO8:FORY=1TO8
406 POKEA+P*Y+X+Q,96
408 NEXT:NEXT
410 GOTO80

```

ROBOTICA

ECCO I FAMOSI KIT "MOVIT" PER
GLI APPASSIONATI DEL "FAI DA TE"

**LINE TRACER II MV-913**

Guidato da un sensore a raggi infrarossi questo robot segue qualsiasi linea nera tracciata su un foglio bianco. L.115.000

**PIPER MOUSE MV-915**

Usando un fischietto unito al kit PIPER MOUSE eseguirà immediatamente i comandi comunicativi: sinistra, stop, destra, stop, avanti e stop. L.89.000

**MEMOCOM CRAWLER MV-918**

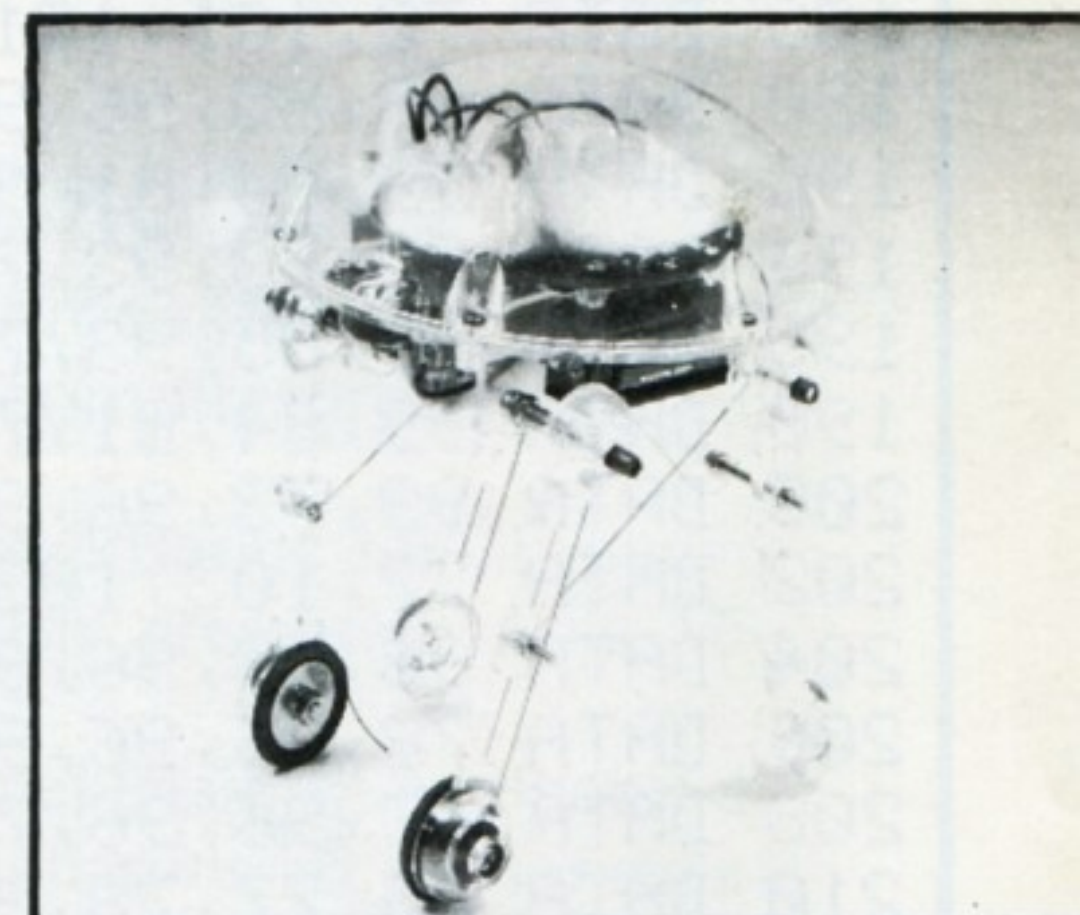
Questo robot è controllato da una memoria RAM 256x4 bit, mediante la tastiera inclusa nel kit e si muove in base ai comandi comunicati a quest'ultima. L.149.000

**CIRCULAR MV-935**

È provvisto di due grandi ruote che gli permettono di ruotare in avanti, a destra, a sinistra e su se stesso. Tutti i movimenti sono controllati mediante un comando a distanza. L.149.000

**NAVIUS MV-938**

È un robot controllato mediante dischetti programmabili. È in grado di riconoscere il bianco e il nero del dischetto programmato e quindi muoversi in avanti, a destra, a sinistra o fermarsi. L.115.000

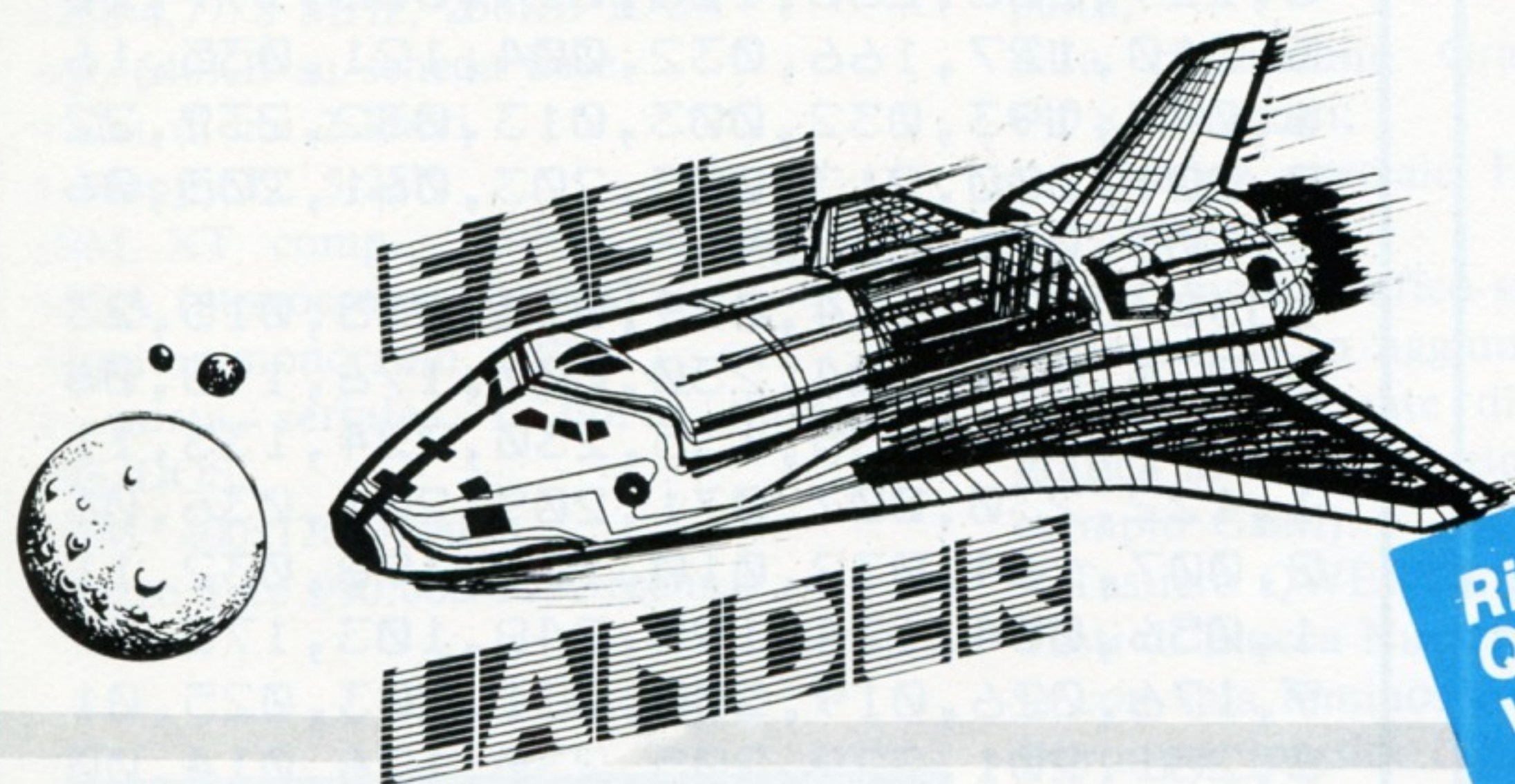
**MEDUSA MV-939**

Il robot dal cervello elettronico. Registrati i comandi, il robot inizierà a muoversi femandosi automaticamente passato il tempo prestabilito. L.65.000

PRODOTTI IN GIAPPONE - DISTRIBUITI IN ITALIA
da **TEXIM ITALIA s.r.l.** - Via Flavio Stilicone, 111
C.A.P. 00175 - Roma - Tel. 7665495

Per riceverli:

Inviare l'importo al detto indirizzo, a mezzo **VAGLIA POSTALE** o **ASSEGNO BANCARIO N.T.** in busta chiusa.



Riuscirete ad atterrare su un pianeta ostile?
Questo è un gioco niente affatto semplice!
Ve ne accorgerete e vi ci appassionerete.

FAST LANDER è un gioco di simulazione. Dovete riuscire a far atterrare la vostra navicella in uno spazio piuttosto esiguo su un pianeta alquanto inospitale.

Il movimento della navetta risulta molto realistico grazie ad una routine in linguaggio macchina, a cui è anche affidato il compito di rilevare l'impatto con il suolo.

Digitate con calma il listato e i codici decimali e poi date RUN 800. Il programma visualizzerà le istruzioni mentre verrà allocato il linguaggio macchina. Ad operazione ultimata incomincerà il gioco con la visualizzazione del terreno roccioso, del modulo spaziale e della piattaforma di atterraggio. I comandi per la guida sono i seguenti:

- tasto 1** accelerazione verso il basso
- tasto 2** bilanciamento della forza di gravità
- tasto 3** accelerazione verso l'alto
- tasto 8** virata a sinistra
- tasto 9** ritorno alla posizione verticale
- tasto 0** virata a destra

Evitate di cambiare i numeri delle linee durante la fase di copiatura del programma in quanto qualche istruzione di solito utilizza dei parametri variabili.

Vi assicuriamo che il gioco non è affatto semplice come potrebbe sembrare a prima vista: bastano un paio di partite per rendersene conto. Se poi siete dei provetti piloti allora potete aumentare il livello di difficoltà dando GOTO 3000 e poi RUN.

LIST

1

```

1 REM *****
2 REM * FAST LANDER *
3 REM * ANGELO COCCETTINI *
4 REM * GIULIO VANNINI *
5 REM * COPYRIGHT LIST 1987*
6 REM *****
7:
9 GO SUB 400
10 GO SUB 100+PEEK 23560
20 LET sx=sx+ax: LET sy=sy+ay:
LET x=x+sx: LET y=y+sy
24 POKE 32300,x: POKE 32301,19
2-y: GO TO USR 32308
147 RETURN
148 LET ax=q: POKE 32302,96: RE
TURN

```

LIST

2

```

149 LET ay=-q: RETURN
150 LET ay=0: RETURN
151 LET ay=q: RETURN
155 RETURN
156 LET ax=-q: POKE 32302,104:
RETURN
157 LET ax=0: POKE 32302,88: RE
TURN
356 RETURN
400 CLS : LET q=.05: RESTORE 42
0: PLOT 0,4: FOR x=0 TO 247 STEP
8: READ y: DRAW 8,y: NEXT x: DR
AW 7,-4: POKE 32304,0: POKE 3230
5,0: POKE 32302,88
410 LET a$=""
FOR i=6 TO 15 STEP 3: PRINT AT
i,0: a$(i+1 TO ):a$( TO i): NEXT
i
412 PLOT 103,3: DRAW 10,0
414 RANDOMIZE : LET x=32+200*RN
D: LET y=192-10*RND: LET sx=-.5*
RND: LET sy=0: LET ax=sy: LET
ay=sy
416 DEF FN t(a)=PEEK (a)+256*PE
EK (a+1)+65576*PEEK (a+2): LET t
0=FN t(23672)
418 RETURN
420 DATA 3,4,-1,-2,-3,-2,1,4,3,
-1,1,-3,-4,0,2,-2,2,3,2,4,-2,1,-
3,-4,-2,2,3,2,1,-1,-4,-3
800 CLEAR 31999: POKE 23675,88:
POKE 23676,127: BORDER 1: INK
7: PAPER 0: CLS : GO SUB 2000:
RESTORE 1500: PRINT : PRINT " A
ttendere: sto creando il L/M."
810 LET loc=32302: LET tot=0
815 READ a: IF a=20000 THEN GO
TO 850
820 LET tot=tot+a
823 IF a>255 AND tot/2<>a THEN
CLS : PRINT FLASH 1;AT 10,10;"

```


LIST

3

```

DATA ERROR": STOP
824 IF a>255 AND tot/2=a THEN
LET tot=0: GO TO 815
825 POKE loc,a: LET loc=loc+1
830 GO TO 815
850 PRINT AT 14,0;"      Premi un
tasto per iniziare "
860 IF INKEY$="" THEN GO TO 86
0
870 CLS : BORDER 1: GO TO 5
1000 PRINT AT 0,0;
1030 IF (sy*sy+sx*sx)>.05 OR y>4
0 THEN POKE 32302,112: RANDOMI
ZE USR 32308: RANDOMIZE USR 3239
3: PRINT FLASH 1;" DISTRUTTO!":
LET score= 200-INT (y): GO TO 1
050
1034 IF (x>106 OR x<103) THEN P
RINT FLASH 1;" MANCATO!": FOR i
=1 TO 4: FOR j=20 TO 12 STEP -2:
BEEP .01,j: NEXT j: NEXT i: PLO
T 107,4: DRAW 0,24: PRINT AT 1,0
: LET score=400-INT (ABS (x-104.
5)): GO TO 1050
1038 BEEP .2,20: BEEP .3,10: LET
t0=INT ((FN t(23672)-t0)/50): P
RINT FLASH 1;"ATTERRATO IN ";t0
;" SEC.": LET score=INT (400+200
/t0): FOR u=1 TO 500: NEXT u: G
O TO 1050:
1050 PRINT "PUNTEGGIO=";score: G
O TO 850
1100 IF y<8 THEN LET y=8
1110 IF y>191 THEN LET y=191
1120 LET sy=-sy: GO TO 24
1200 IF x<0 THEN LET x=0
1210 IF x>247 THEN LET x=247
1220 LET sx=-sx: GO TO 24
1500 DATA 088,127,160,160,088,12
7,042,048,126,237,091,050,126,20
5,061,127,042,044,126,001,076,00
4,124,214,184,208,001,176,004,12
5,214,248,208,034,048,126,237,09
1,046,126,237,083,050,126,213,20
5,204,126,209,001,010,000,040,00
3,001,232,003,197,213,229,033,19
2,064,205,07446
1510 DATA 176,126,033,128,072,20
5,176,126,033,032,072,205,156,12
6,033,224,072,205,156,126,225,20
9,205,038,127,193,201,014,000,03
3,196,126,006,000,016,254,126,21
1,254,035,013,032,245,201,000,00
0,017,032,000,014,008,126,025,02
3,006,032,043,203,022,016,251,03
6,013,032,06341
1520 DATA 242,201,125,246,031,09
5,014,008,006,032,084,026,031,20

```

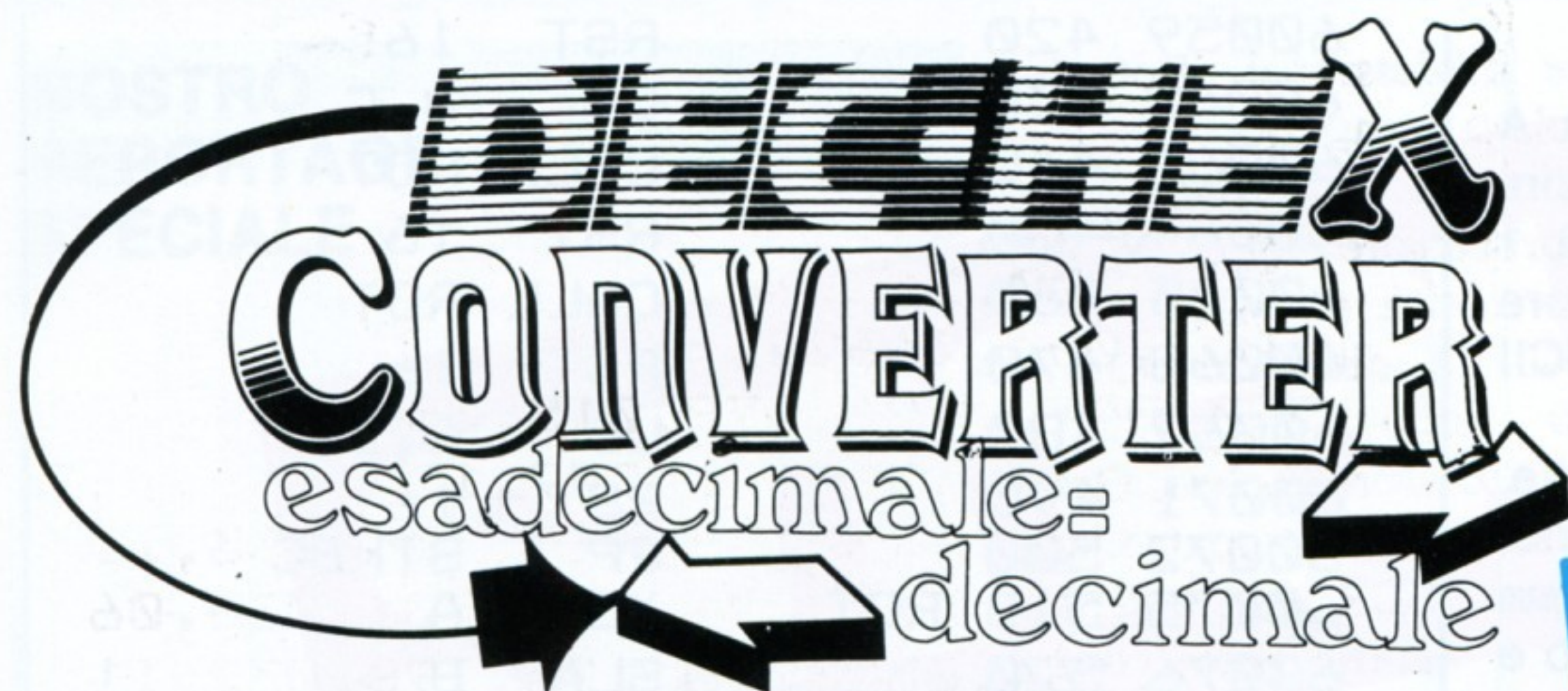
LIST

4

```

3,030,035,016,251,125,214,032,11
1,036,013,032,238,201,000,000,00
0,229,205,230,126,014,008,197,20
5,010,127,166,032,004,121,035,16
6,043,193,032,003,013,032,239,22
5,201,000,213,085,203,061,203,06
1,203,061,06624
1530 DATA 124,079,015,015,015,23
0,024,071,124,230,007,176,198,06
4,103,121,135,135,230,224,133,11
1,122,230,007,071,209,201,036,06
2,007,164,032,010,125,198,032,11
1,056,004,124,198,248,103,175,07
9,176,026,019,200,031,203,025,01
6,251,201,229,205,230,126,014,00
8,197,205,07530
1540 DATA 010,127,182,119,035,12
6,177,119,043,193,013,032,241,22
5,201,229,205,230,126,014,008,19
7,205,010,127,238,255,166,119,03
5,121,238,255,166,119,043,193,01
3,032,237,225,201,000,060,126,12
6,060,060,066,195,028,062,126,12
6,062,068,194,003,056,124,126,12
6,124,034,07802
1550 DATA 067,192,036,129,090,06
0,060,090,129,036,001,015,031,06
3,063,063,031,007,240,248,252,25
2,252,248,248,048,024,124,124,06
0,009,097,240,096,000,028,255,25
5,255,254,252,012,000,000,192,24
8,112,000,003,003,000,000,000,00
0,000,000,000,000,000,000,000,00
0,000,000,05594
1560 DATA 20000
2000 PRINT "          * FAST LANDE
R *"
2005 PRINT
2010 PRINT "[1] accelerazione in
basso"
2012 PRINT
2013 PRINT "[2] bilanciamento fo
rza gravita'"
2015 PRINT
2020 PRINT "[3] accelerazione in
alto"
2025 PRINT
2030 PRINT "[8] virata a sinistr
a"
2035 PRINT
2040 PRINT "[9] ritorno posizion
e verticale"
2045 PRINT
2050 PRINT "[0] virata a destra"
2060 RETURN
3000 POKE 32442,0: POKE 32443,23
7: POKE 32444,103: POKE 32419,0:
POKE 32423,237: POKE 32424,111

```

Come convertire un numero decimale in esadecimale usando il linguaggio macchina del vostro ZX-SPECTRUM.

Introduzione

Tutti i possessori di Spectrum avranno notato la solita cronica lentezza dell'interprete BASIC. Questa lentezza del BASIC limita considerevolmente il campo di applicazione del computer in quei settori dove è proprio la velocità la base essenziale dell'intero programma, come per esempio nei videogiochi. Qualcuno potrebbe puntualizzare che in definitiva gli Arcades non sono l'applicazione più seria per un computer e che esistono dei validissimi programmi in BASIC di gestione, di studio, ecc. Le affermazioni di tali utenti sono perfettamente logiche e in gran parte condivise da tutti noi. È chiaro comunque che la scarsa velocità dell'interprete ci toglie la possibilità di scrivere un bel programma di animazione che dà senz'altro molte più soddisfazioni di uno di gestione. Per questo motivo da questo numero inizieremo la pubblicazione di una serie di routine in linguaggio macchina molto utili sia dal punto di vista applicativo che da quello didattico. Infatti oltre ai normali codici nelle solite linee data verrà proposto anche il listato in ASSEMBLER completo di commenti. In questo modo speriamo di avvicinare l'utente Spectrum ad un linguaggio che permette operazioni decisamente impossibili al BASIC, fornendo allo stesso tempo alcune piccole routines che si riveleranno molto utili durante la stesura dei programmi.

Dec-Hex Converter

La prima routine che incontriamo converte un numero decimale N compreso tra 0 e 65535 nella sua rappresentazione esadecimale stampando il risultato sullo schermo. Per utilizzare la routine non servono strani POKE per passare il numero alla routine; infatti basta digitare in modo diretto o da programma

RANDOMIZE N + USR 60000

La routine chiamata con la USR preleva dallo STACK del calcolatore il numero N e lo traduce in esadecimale, provvedendo a visualizzarne il formato sullo schermo. Così, se battete

RANDOMIZE 65518 + USR 60000

otterrete

HEX(65518) = FFEE

Commento al Listato Assembler

Linee 40-90

Definizione delle costanti numeriche per l'ASSEMBLER.

Linee 100-180

Si controlla se il numero N è stato passato al L/M con la istruzione RANDOMIZE N od una equivalente (PRINT N + USR 60000, LET a=N + usr 60000, ecc). Ciò viene fatto sottraendo all'indirizzo memorizzato nella variabile di sistema STED, che punta la fine dello STACK del calcolatore, il numero STBT, che ne definisce l'inizio. Se non si ottiene il numero cinque, equivalente allo spazio occupato da un numero nella rappresentazione in Floating Point, viene generato un messaggio di errore con l'istruzione RST 8 seguita dal codice dell'errore-1.

Linee 190 e 200

Si carica l'accumulatore con il numero due e si apre il canale associato allo schermo, che è appunto quello #2, chiamando una subroutine nella ROM.

Linee 210-230

Altre chiamate alla ROM. Si preleva il numero N dallo STACK del calcolatore chiamando UNSTK, che trasporta il numero nei registri BC generando un messaggio di errore se $N > 65535$. Si salva temporaneamente N sullo STACK del linguaggio macchina (che da questo momento chiameremo semplicemente stack) e ripristina il valore sullo STACK del calcolatore (si badi bene che lo STACK del calcolatore non coincide con lo stack).

Linee 240-360

Viene stampato il messaggio HEX(N)= con una serie di chiamate alla ROM (RST 16 con l'accumulatore A che contiene il codice da stampare). Il numero viene stampato dalla ROM-routine PRSTK.

Linee 380-490

Il numero N supposto a 16 BIT viene diviso in quattro numeri da 4 BIT; ognuno di questi valori è tradotto nella rappresentazione esadecimale e poi stampato da queste istruzioni.

Linea 500

Viene introdotto sullo stack un valore qualsiasi per evitare che il sistema entri in errore e si ritorna al sistema in modo indiretto tramite STKBC.



Linea 510-640

Questa subroutine in ordine 4 BIT alla volta da B a A usando le istruzioni di SHIFT (RLA e SLA). Dopodiché converte A nella rappresentazione ASCII. Così 0 diventa 48, 1 si trasforma in 49 ecc. Se si ottiene un numero maggiore di 57 si aggiunge 7 per trasformare 10 in A (codice ASCII 65), 11 in B e così via.

Un piccolo avviso. La routine così scritta non è rilocabile, cioè non può essere trasportata in un'altra parte in memoria e pretendere che funzioni. Chi possiede comunque un buon ASSEMBLER può tranquillamente copiare il listato e cambiare la ORG alla linea 30. Noi abbiamo utilizzato l'ottimo assembler della HISOFT, GENS2. È chiaro comunque ci si può avvalere di un qualsiasi altro programma che sia in grado di compilare un file ASSEMBLER. Chi non disponesse di nessun tipo di compilatore può utilizzare il programma in BASIC per la creazione di DEC-HEX CONVERTER badando al fatto che la routine ottenuta in questo modo non risulta rilocabile.

```

600000 30      ORG 600000
5633 40 CHANOP EQU 5633
7833 50 UNSTK EQU 7833
11563 60 STKBC EQU 11563
23653 70 STED EQU 23653
23651 80 STBT EQU 23651
8242 90 PRSTK EQU 8242
600000 100     LD HL,(STED)
600003 110     LD DE,(STBT)
600007 120     AND A
600008 130     SBC HL,DE
600010 140     LD A,L
600011 150     CP 5
600013 160     JR NC,CONT
600015 170     RST 8
600016 180     DEFB 25
600017 190 CONT LD A,2
600019 200     CALL CHANOP
600022 210     CALL UNSTK
600025 220     PUSH BC
600026 230     CALL STKBC
600029 240     LD A,"H"
600031 250     RST 16
600032 260     LD A,"E"
600034 270     RST 16
600035 280     LD A,"X"
600037 290     RST 16
600038 300     LD A,"("
600040 310     RST 16
600041 320     CALL PRSTK
600044 330     LD A,")"
600046 340     RST 16
600047 350     LD A,"="
600049 360     RST 16
600050 370     POP BC
600051 380     XOR A
600052 390     CALL ROT
600055 400     RST 16
600056 410     CALL ROT

```

```

600059 420     RST 16
600060 430     LD B,C
600061 440     CALL ROT
600064 450     RST 16
600065 460     CALL ROT
600068 470     RST 16
600069 480     LD A,13
600071 490     RST 16
600072 500     JP STKBC
600075 510 ROT  XOR A
600076 520     SLA B
600078 530     RLA
600079 540     SLA B
600081 550     RLA
600082 560     SLA B
600084 570     RLA
600085 580     SLA B
600087 590     RLA
600088 600     ADD A,48
600090 610     CP 58
600092 620     RET C
600093 630     ADD A,7
600095 640     RET

```

```

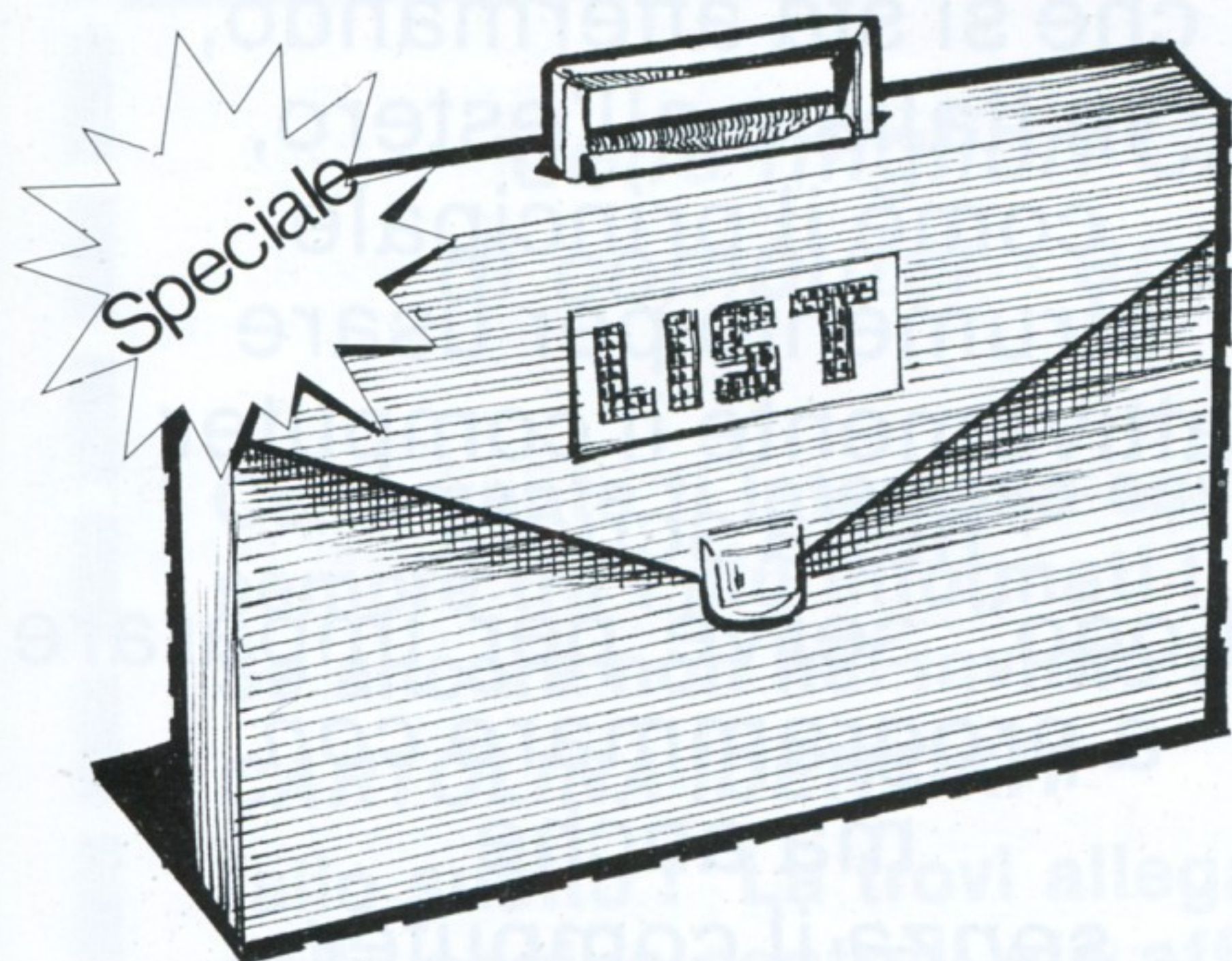
1 REM * DEC-HEX CONVERTER *
2:
10 DATA 042,101,092,237,091,09
9,092,167,237,082,125,254,005,04
8,002,207,025,062,002,205,001,02
2,205,153,030,197,205,043,045,06
2,072,215,062,069,215,062,088,21
5,062,040,215,205,050,032,062,04
1,215,062,061,215,193,175,205,17
1,234,215,205,171,234,215,065,20
5,171,234,08089
20 DATA 215,205,171,234,215,06
2,013,215,195,043,045,175,203,03
2,023,203,032,023,203,032,023,20
3,032,023,198,048,254,058,216,19
8,007,201,000,000,000,000,000,00
0,000,000,000,000,000,000,000,00
0,000,000,000,000,000,000,000,00
0,000,000,04000
30 DATA 20000
70 RESTORE : LET tot=0
80 INPUT "start: ";loc
90 READ a: IF a=20000 THEN ST
OP
100 LET tot=tot+a
110 IF a>255 AND tot/2<>a THEN
CLS : PRINT FLASH 1;AT 10,10;"
DATA ERROR": STOP
120 IF a>255 AND tot/2=a THEN
LET tot=0: GO TO 90
130 POKE loc,a: PRINT loc,a
140 LET loc=loc+1
150 GO TO 90

```




Le Raccolte di LIST 1984-1985

UNA VERA SOFTECA CON 364 PROGRAMMI
PREZIOSE RUBRICHE DI CULTURA INFORMATICA
E MOLTI CONSIGLI... OLTRE I MANUALI



IN REGALO

LA VALIGETTA DI MODA

Per richiederle usa il bollettino qui a fianco
su cui è stampato l'importo da inviare.

CONTI CORRENTI POSTALI
RICEVUTA di un versamento di L.

Lire

72609001
sul C/C N. **72609001**
intestato a **LIST - Programmi per il tuo home computer**
computer Via Flavio Stilicone, 111 - 00175 ROMA
eseguito da _____
residente in _____ via _____ addi _____

CONTI CORRENTI POSTALI
Certificato di accredittam. di L.

Lire

72609001
sul C/C N. **72609001**
intestato a **LIST - Programmi per il tuo home computer**
computer Via Flavio Stilicone, 111 - 00175 ROMA
eseguito da _____
residente in _____ via _____ addi _____

Bollo lineare dell'Ufficio accettante

numerato d'accettazione

Cartellino del bollettario

Bollo a data

Bollo lineare dell'Ufficio accettante

L'UFF POSTALE

Cartellino del bollettario

Bollo a data

Bollo a data

Bollo a data

data

data

progress.

progress.

numero conto

numero conto

importo

importo

Importante: non scrivere nella zona sottostante!

N. del bollettario **ch 9**

Mod ch-8-b/s AUT (1984) Cod - 127902

IMPORTANTE: non scrivere nella zona soprastante!

AVVERTENZE

Per eseguire il versamento, il versante deve compilare in tutte le sue parti, a macchina o a mano, purché con inchiostro nero o nero-blauastro il presente bollettino (indicando con chiarezza il numero e la intestazione del conto ricevente qualora già non siano impressi a stampa).

NON SONO AMMESSI BOLLETTINI RECANTI CANCELLATURE, ABRASIONI O CORREZIONI.

A tergo del certificato di accredito i versanti possono scrivere brevi comunicazioni all'indirizzo dei correntisti destinatari.

La ricevuta non è valida se non porta i bolli e gli estremi di accettazione impressi dall'Ufficio postale accettante.

La ricevuta del versamento in Conto Corrente Postale, in tutti i casi in cui tale sistema di pagamento è ammesso, ha valore liberatorio per la somma pagata con effetto dalla data in cui il versamento è stato eseguito.

1 P.Z. OFFICINA D.V. ROMA

TAGLIANDO
questo tagliando ricevuta
costituisce documento idoneo e
sufficiente ad ogni effetto contabile
in quanto l'IVA sui periodici è
assolta dall'editore
NON SI RILASCIANO FATTURE

Spazio per la causale del versamento

- ☐ Raccolta LIST 84 c/cartella L. 25.000
comprese L. 4.000 sp. sped.
- ☐ Raccolta LIST 85 c/cartella L. 35.500
comprese L. 4.000 sp. sped.
- ☐ Libro: io parlo LOGO L. 24.500
comprese L. 1.500 sp. sped.
- ☐ Abbonamento a 11 numeri di LIST L. 55.000

SI PREGA DI COMPILARE IN STAMPATELLO

Parte riservata all'Ufficio dei Conti Correnti



INDISPENSABILE
AI NEO-COMPUTERISTI



FINALMENTE IL PRIMO LIBRO TUTTO ITALIANO SUL “LOGO”

Il “Logo” è un linguaggio
che si sta affermando,
in Italia e all'estero,
come il principale
strumento per usare
attivamente il computer
nella scuola.

Il “Logo” serve per imparare
a programmare *con*,
ma anche
senza il computer.

Per richiederlo usa il bollettino

SCUOLA & COMPUTER

IL COMPUTER e L'UGUAGLIANZA

Il lettore si rassicuri, con questa nota non ci vogliamo occupare di problemi etico-giuridici o etico-politici sulla considerazione di «uguaglianza» dei membri di una collettività relativamente ad alcuni fatti o valori, né di una ipotetica «maggiore uguaglianza» dei computeristi, rispetto agli altri membri della società (insomma, chi sa usare un computer, non è, per questo «più uguale» degli altri).

ci occupiamo di come il computer tratta l'uguaglianza, perché, come vedremo, ha un modo tutto suo di considerarla.

1. Tutti sanno che l'uguaglianza gode (si fa per dire) della proprietà riflessiva, simmetrica e transitiva.

Per la *proprietà riflessiva*, ogni oggetto A (matematico e non) è uguale a se stesso: **A = A**;

per la *proprietà simmetrica*, se un oggetto A è uguale ad un oggetto B allora anche B è uguale ad A:

se A = B allora B = A;

per la *proprietà transitiva*, se A è uguale a B e B è uguale a C, allora A è uguale a C:

se (A = B e B = C) allora A = C.

Prescindiamo dalla prima proprietà, verso la quale il computer, come si

dice, non fa una piega, e prendiamo in considerazione la seconda, la proprietà riflessiva.

Scriviamo, dunque, i due seguenti programmi:

```
10 LET A=5
```

```
20 LET B=A
```

```
30 PRINT B
```

```
10 LET A=5
```

```
20 LET A=B
```

```
30 PRINT B
```

(in molti computer l'istruzione LET si può omettere).

Sulla base della proprietà simmetrica, producendo $A=B$ e $B=A$ lo stesso effetto, i due programmi dovrebbero dare come risultato 5.

Invece, provare per credere, il primo programma dà per risultato 5, come ci attendevamo, ed il secondo dà 0.

Che cosa è successo?

Semplicemente che il computer ignora la proprietà simmetrica e, mentre nel primo programma riceve l'ordine di dare a B lo stesso valore di A, che è 5, risponde correttamente; nel secondo programma, invece, l'istruzione della linea 20 ordina al computer di dare ad A il valore di B. Ma non essendo stato assegnato a B alcun valore, per il computer B è uguale a 0 e pertanto stampa questo valore.

Passiamo, adesso, alla *proprietà transitiva*, e scriviamo il programma:

```
10 LET A=5
```

```
20 LET A=B
```

```
30 LET B=C
```

```
30 PRINT C
```

Se lo facciamo «girare» otteniamo per C il valore 0.

Però, ormai sappiamo che la linea 20 e 30 sono errate perché assegnano ad A e a B valori non ancora specificati. Correggiamo, dunque, in:

```
10 LET A=5
```

```
20 LET B=A
```

```
30 LET C=B
```

```
40 PRINT C
```

Questa volta otteniamo per C il valore 5 cioè il valore di A.

La proprietà transitiva è dunque valida. Però, non essendo valida la proprietà simmetrica, deve essere riscritta così:

se (B = A e C = B) allora C = A.

2. A tutti sarà capitato di rimanere perplessi nel leggere, la prima volta che in informatica si sono imbattuti in «uguaglianze» numeriche come $N = N + 1$, oppure, trattandosi di stringhe, come $A\$ = A\$ + B\$$, perché sono evidentemente uguaglianze errate. Il fatto è che queste non sono uguaglianze ma... istruzioni, che si danno al computer per sostituire il valore N con il valore $N + 1$, oppure la stringa A\$ con la stringa A\$ seguita dalla stringa B\$. Se, ad esempio:

$A\$ = \text{"Ciao"}$ e $B\$ = \text{"a tutti"}$,
 $A\$ = A\$ + B\$$ significa sostituire "Ciao" con "Ciao a tutti".

GIEFFE

NUMERI PRIMI

Per tutti
i COMPUTER

Tutti sanno che i numeri naturali sono divisibili per se stessi e per l'unità:

3 è divisibile per 3 e per 1,

36 è divisibile per 36 e per 1, ...

Vi sono, però, alcuni numeri, come 2, 3, 5, 7, ... che sono divisibili soltanto per se stessi e per l'unità e che sono detti "numeri primi". I numeri non primi si chiamano "numeri composti".

Ad esempio, 4 è un numero composto perché, oltre ad essere divisibile per 4 e per 1 è divisibile anche per 2. La successione dei numeri primi ha questo inizio:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, ...

come si comprende, 2 è l'unico numero primo pari, perché qualunque altro numero pari è divisibile, oltre che per se stesso e per l'unità, certamente anche per 2.

I numeri primi hanno suscitato notevole interesse fin dai tempi più antichi: la successione è limitata o illimitata? di coppie di numeri come 5 e 7, 11 e 13, 17 e 19 (i cosiddetti numeri primi gemelli) ce ne sono in numero finito o infinito? è possibile trovare una formula che dia tutti (e soli) i numeri primi o i numeri gemelli?, ...

La risposta alla prima domanda fu data da Euclide, verso l'anno 300 a.C., nei suoi «Elementi» (libro IX, proposizione 20), con una dimostrazione elementare, rimasta un modello di ragionamento matematico.

Euclide ragionò così: supponiamo che ad un certo punto la successione dei numeri primi abbia termine e sia P_n l'ultimo numero primo.

sia P il numero ottenuto moltiplicando tutti i numeri primi, da P_1 a P_n , ed aumentando questo prodotto di 1:

$$P = P_1 \cdot P_2 \cdot P_3 \cdot \dots \cdot P_n + 1$$

Questo numero P o è primo o non lo è. Se fosse primo, allora c'è almeno un altro numero primo oltre quelli ammessi, il che è contrario all'ipotesi che i numeri P_1, P_2, \dots, P_n , siano TUTTI i numeri primi.

Se non è primo, allora è composto e perciò dovrebbe essere divisibile almeno per uno dei numeri primi da P_1 a P_n . Ma P , diviso per P_1 , per P_2, \dots , per P_n , dà sempre resto 1, e, dunque, non è divisibile. E allora ci deve essere un altro numero primo, diverso dai dati P_1, P_2, \dots, P_n , che è divisore di P , il che è sempre contrario all'ipotesi ammessa.

Sulla domanda sui numeri primi ge-

melli non c'è risposta, sulla terza, c'è stata una risposta negativa.

Ovviamente, per la successione dei numeri primi si possono inventare delle formule che danno alcuni numeri primi, come ad esempio S_n 1, per $n = 1, 2, 3, \dots$, che fornisce i numeri: **5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, ...**, però, come si vede, assieme ad alcuni numeri primi, ci sono anche 25 e 35 che non lo sono.

Altre formule note sono:

$n^2 + n + 41$ e l'analoga $n^2 - n + 41$, che forniscono i numeri primi

41, 43, 47, 53, ...

ma che cadono in difetto per $n = 41$, avendosi:

$$\begin{aligned} n^2 + n + 41 &= 41^2 + 41 + 41 = \\ &= 41(41 + 2) = 41 \cdot 43 \end{aligned}$$

e

$$n^2 - n + 41 = 41^2 - 41 + 41 = 41^2$$

Quasi tutti i ragazzi che sono venuti a conoscenza dei problemi lasciati aperti dai numeri primi, si sono cimentati, ma inutilmente, per trovare, magari durante l'intervallo per la merenda, una formula che, se non tutti i numeri primi, ne desse almeno un bel po'.

Come è noto, per scrivere tutti i numeri primi, sino ad un certo numero, si può usare un metodo empirico il "cricello di Eratostene", e cioè:

a) si scrive il 2 seguito da tutti i numeri dispari;

b) lasciato il 3, si cancellano tutti i suoi multipli 6, 9, 12, ...;

c) si continua così cancellando i multipli di 5 (10, 15, 20, ...), i multipli di 7 (14, 21, 28, ...) ecc.

Ma noi, ovviamente, per la ricerca dei numeri primi, ci serviremo del computer. Battiamo, dunque, il programma riportato nella pagina che segue (valido per qualsiasi computer), che, a parte il 2 e il 3, ricerca i numeri primi tra tutti i numeri dispari che non sono divisibili per nessun numero (oltre, ovviamente, se stessi e l'unità).

Vogliamo vedere quanto tempo impiega per scrivere i numeri primi minori di 100?

Inseriamo l'orologio nel programma al momento dell'inizio della ricerca, e cioè prima della linea 50: (solo per Commodore 64, per gli altri computer, consultare il manuale).

45 TI\$ = "000000"

e ne facciamo stampare il valore non appena è finita la ricerca, cioè alla linea 160. Modifichiamo, perciò, il pro-

gramma da questa linea, per inserire la stampa dell'orologio:

160 PRINT:PRINT:PRINT:"RICERCA EFFETTUATA IN:"

170 PRINT LEFT\$(TI\$,2) "ORE "MID\$(TI\$,3,2) " MIN "RIGHT\$(TI\$,2) " SEC."

Dato il RUN, possiamo constatare che la ricerca avviene in 38 secondi. Possiamo, però, migliorare il programma, rendendolo più veloce, osservando che non è necessario eseguire le divisioni fino ad $N-1$, perché basta arrestarsi alla radice quadrata di N .

Se, infatti, consideriamo un numero qualsiasi, ad esempio 36, vediamo che le coppie dei numeri che, moltiplicati, danno 36, ad un certo punto si ripetono, in ordine inverso:

1*36, 2*18, 3*12, 4*9, 6*6, 8*4, 12*3, 18*2, 36*1

Modifichiamo, dunque, la linea 130, sostituendo a $N-1$ la radice quadrata di N . Anzi, poiché non è detto che la radice quadrata di N sia un numero naturale, considereremo solo la parte intera di tale radice quadrata:

130 IF D > INT(SQR(N)) THEN PRINTN";GOTO30

Vediamo, adesso, che il tempo di ricerca si è ridotto a soli 14 secondi. Se dovessimo compilare la tavola dei numeri primi, mettiamo, fino a 10.000, il programma ci sembrerà molto lento, perché via via che avanza nella ricerca, il numero delle divisioni da provare, aumenta sempre più.

Possiamo cercare di guadagnare ancora qualche cosa "compattando" il programma, cioè togliendo le REM, scrivendo in ciascuna linea tutte le istruzioni possibili, e rinumerando il programma a partire da 1 con passo 1, così:

1 PRINT:PRINT:"SUCESSIONE DEI NUMERI PRIMI MINORI DI 100":
ti\$ = "000000"

2 PRINT"2, 3, ":N = 3

3 N = N + 2:IFN > 99THEN9

4 D = 2

5 D = D + 1

6 IFN/D = INT(N/D)THEN3

7 IFD > INT(SQR(N))THENPRINTN";GOTO3

8 GOTO5

9 PRINT:PRINT:PRINT:"RICERCA EFFETTUATA IN:"

10 PRINTLEFT\$(TI\$,2) "ORE "MID\$(TI\$,3,2) " MIN "RIGHT\$(TI\$,2) " SEC."

Possiamo osservare che, adesso, la ricerca si è ridotta ancora di 2 secondi.

Se, però misuriamo il tempo via via che operiamo le riduzioni del listato, vediamo che la riduzione a 12 secondi si ottiene appena togliamo le REM. Togliendo gli spazi e rinumerando il listato, non si ottiene nessun'altra riduzione.

Proviamo, allora a scrivere i numeri primi minori di 1000, modificando opportunamente la linea 90.

il programma completo li calcola in **6 min e 31 sec**, quello senza le REM in **5 min e 46 sec**, il programma compatto in **5 min e 41 sec**, quello rinumerato in **5 min e 40 sec**.

Come si vede il risparmio maggiore di tempo si ha sopprimendo le REM. La compattazione e la rinumerazione producono poco effetto perché il programma è breve. Per i programmi lunghi si ottiene un effetto migliore, ma sempre relativo. Se il vostro computer non rinumerava automaticamente le li-

nee o se non avete un programma manuale, risparmiatemi pure la fatica: ciò che si guadagna in memoria e in velocità non vale il lavoro fatto per la rinumerazione.

Per rendere notevolmente più veloce un programma occorre scriverlo in linguaggio macchina o, almeno, "compilarlo".

```

10 REM *****
20 REM *          SUCCESSIONE DEI NUMERI PRIMI          *
30 REM *          LIST - SCUOLA & COMPUTER              *
40 REM *****
50 PRINT:PRINT"SUCCESSIONE DEI NUMERI PRIMI MINORI DI 100"
60 PRINT "2, 3,";
70 N=3:          REM PRIMO NUMERO DISPARI CONSIDERATO
80 N=N+2:        REM SUCCESSIVI NUMERI DISPARI
90 IF N>99 THEN 160: REM SE IL NUMERO SUPERA 99 LA RICERCA
    HA FINE
100 D=2:          REM PRIMO DIVISORE
110 D=D+1:        REM DIVISORI SUCCESSIVI
120 IF N/D=INT(N/D) THEN 80: REM TROVATO UN DIVISORE PASSA AL
    NUMERO SUCCESSIVO
140 REM ARRIVATO ALL'ULTIMA PROVA STAMPA N, CHE È PRIMO
150 GOTO 110:     REM PASSA AL DIVISORE SUCCESSIVO
160 END

```

Ci sono, infatti, in commercio dei "compilatori", come l'Austro Speed per il C64, che traducono automaticamente il programma in un linguaggio molto vicino a quello macchina.

L'ultimo programma, quello compatto e rinumerato, se è anche compilato, calcola i numeri primi fino a 1000 in 4 min e 14 sec. □

INFORMATICA E DIDATTICA

Due aspetti fondamentali

di Elisabetta Portacci

Se osserviamo attentamente possiamo notare la graduale creazione di **due aspetti** all'interno del rapporto fra informatica e didattica: da una parte le applicazioni del **computer come strumento didattico**, quelle applicazioni, cioè, in cui il computer viene programmato per interagire con l'allievo ed essere, a seconda dei casi, docente, esaminatore, aiutante o partner di scoperta e di gioco; dall'altra il computer, e più in generale, **l'informatica come oggetto d'insegnamento** nella formazione scolastica non specialistica.

Gradualmente ci si è resi conto che si tratta di due aspetti certamente distinti, ma di fatto non separabili, almeno per quanto riguarda la scuola. Infatti, quando il computer entra nelle classi, insegnanti e studenti, approfondendo la loro conoscenza dell'informatica, sono portati, in modo naturale, ad applicarla al loro specifico campo d'attività, **l'insegnamento-apprendimento**: avviene, cioè, anche nella didattica, quello che sta avvenendo in svariati altri settori: il non specialista, accostandosi all'informatica, ne scopre le potenzialità in relazione alla propria professione e, gradualmente, diventa attore di un processo di rinnovamento metodologico.

Altrettanto improponibile sarebbe utilizzare nelle scuole il computer come mezzo didattico mantenendo un velo di mistero sulle nozioni fondamentali dell'informatica, le modalità di funzionamento del computer, le sue potenzialità, in una parola senza ridimensionare il computer da «cervello elettronico» a strumento di lavoro comprensibile, prevedibile e manipolabile da tutti. Non sarebbe educativo, né culturalmente corretto e, probabilmente, non sarebbe accettato.

Esiste poi una terza considerazione, di natura più profonda, che

mette in luce una vera e propria continuità concettuale tra questi due aspetti: in entrambi l'interazione con il computer richiede che l'allievo in qualche modo programmi. Programmare non è soltanto utilizzare il Basic, il Logo o il Pascal o un qualunque altro linguaggio di programmazione per risolvere uno specifico problema. Quasi tutto il miglior software didattico richiede in qualche modo agli allievi di predisporre una serie di operazioni per ottenere un particolare risultato, cioè di programmare. Nel caso di un linguaggio di programmazione, l'allievo deve costruire una sequenza di istruzioni scelte fra quelle appartenenti al linguaggio che sta utilizzando; nel caso del software didattico, deve scegliere fra le operazioni permesse nell'ambiente didattico che il software mette a disposizione. La differenza è solo nel livello delle operazioni, di solito più elementari e meno specifiche nel caso del linguaggio di programmazione, più complesse ed orientate ad un particolare contesto nel caso del software didattico. Fanno eccezione soltanto i programmi didattici più direttivi, che utilizzano strategie analoghe all'istruzione programmata e che non lasciano all'allievo altra possibilità che seguire un percorso predefinito.

Possiamo, quindi, concludere affermando che, molti insuccessi didattici del passato, in alcuni casi un passato non lontano, hanno messo chiaramente in evidenza che la tecnologia è vantaggiosa solo quando fa da supporto a scelte pedagogiche corrette, ad una rigorosa progettazione didattica e ad una buona dose di fantasia e creatività.

Nota: Consultazione di studi e ricerche condotte dall'Istituto di Tecnologie didattiche del C.N.R. di Genova.

di Fabio Russo

Ecco un possibile itinerario didattico attraverso il quale gli alunni possono familiarizzare con concetti informatici relativi a situazioni di gioco e di vita di tutti i giorni

Cito alcuni obiettivi raggiungibili attraverso questo lavoro in ambito cognitivo e affettivo:

- orientarsi nel piano grafico;
- eseguire ordinatamente in senso logico una serie di operazioni;
- esercitare l'analisi nel processo di suddivisione dei Problemi in singole parti;
- sviluppare e perfezionare la capacità di risoluzione dei Problemi;
- incrementare la comunicazione e partecipazione di idee tra gli alunni.

Questo tipo di lavoro si potrebbe proporre ai ragazzi degli ultimi due anni della scuola elementare o di prima media.

I contenuti informatici sono:

- concetto di sequenza logica;
- codici di informazione;
- concetto di stato;
- esecuzione condizionata di operazioni;
- memorizzazione sotto un nome di una serie di operazioni da compiere;
- concetto di automa.

I contenuti matematici sono:

- corrispondenza tra simbolo e oggetto;
- direzione, verso, rotazione;
- individuazione di coordinate rispetto gli assi cartesiani.

Il metodo che si utilizza parte da situazioni di gioco con realizzazioni grafiche di percorsi da compiere, l'uso dei simboli per rappresentare oggetti e uso di procedure.

I mezzi da utilizzare sono dei fogli di carta da lucido (trasparente), lo stradario "TUTTO CITTÀ" della SEAT, alcuni trasferibili per disegno.

All'inizio ogni alunno deve realizzare una griglia su carta da lucido, ricalcandola semplicemente dallo stradario SEAT, e deve imparare a ricercare una via richiesta dall'insegnante, tramite le coordinate (fig. 1).

L'insegnante, una volta assicuratosi che tutti sono in grado di eseguire la ricerca, fa suddividere ulteriormente in spazi più piccoli la griglia (fig. 2).

Le linee verticali della griglia rappresentano le varie vie (colonne), mentre quelle orizzontali indicano i corsi (righe). Ogni alunno diventa un provetto conduttore di taxi che deve svolgere il proprio lavoro nella città e deve rispettare alcune precise regole.

Inoltre deve sapere comunicare con il suo mezzo solamente con alcune parole che conosce e che sa trasformare in azioni.

Due concetti legati alla robotica cominciano a venir fuori, il primo è quello del RISOLUTORE di problemi, nel nostro caso il conduttore; il secondo è quello di ESECUTORE di comandi cioè di taxi.

La griglia è composta da righe e colonne, esse per noi

rappresentano le strade, quindi, vi sono tantissimi pericoli come alcuni incroci semaforizzati, mentre in altri potrebbe casualmente presentarsi un ostacolo.

Vediamo per ora le prime istruzioni che il mezzo è in grado di eseguire, in un secondo tempo ne saranno introdotte altre:

AVANTI: il mezzo avanza di un solo piccolo elemento della griglia. Se incontra un ostacolo non previsto, il motore si spegne;

DESTRA: il mezzo ruota a destra di 90° gradi;

SINISTRA: il mezzo ruota a sinistra di 90° gradi;

INIZIO: il mezzo deve accendere il motore per partire;

FINE: il mezzo non è più in grado di muoversi;

PRENDI: accoglie un passeggero nel taxi;

LASCIA: lo lascia all'incrocio richiesto;

POSTAZIONE: inizialmente il taxi parte sempre dalle coordinate di colonna 1 e riga 1.

Ora si simula la chiamata al radiotelefono del primo taxi che per esempio deve andare a prendere un passeggero nel traffico caotico di Roma, in Piazza A. Capponi e portarlo in Piazza Cavour.

La prima operazione che l'alunno deve compiere è la ricerca della pagina e delle coordinate sullo stradario.

A pagina 21 le coordinate della prima Piazza sono A-5 quelle della seconda C-5.

Si sovrappone alla piantina il lucido con la griglia e si inizia a collocare il simbolo del taxi nella POSTAZIONE di coordinate 1,1 e quello del passeggero in Piazza A. Capponi.

I simboli utilizzati sono rappresentati nella figura 3.

Si studia a questo punto il percorso più agevole (fig. 4). Una volta individuato, si iniziano a dare i comandi che il mezzo conosce, come segue:

Piazza Cavour

INIZIO
DESTRA
AVANTI
AVANTI
AVANTI
SINISTRA
AVANTI
PRENDI
AVANTI
DESTRA
AVANTI
AVANTI
AVANTI
AVANTI
AVANTI
AVANTI
LASCIA
FINE

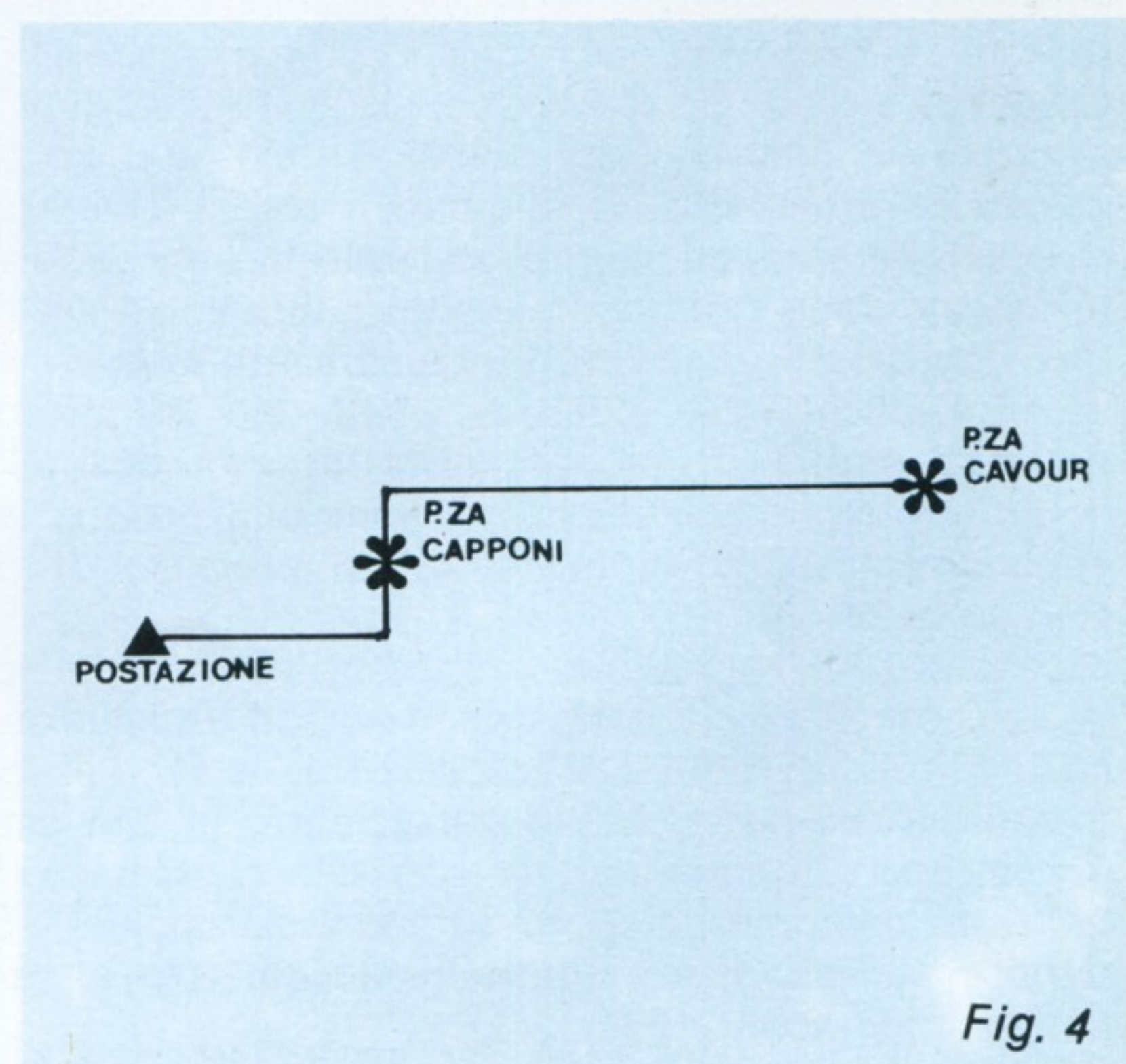


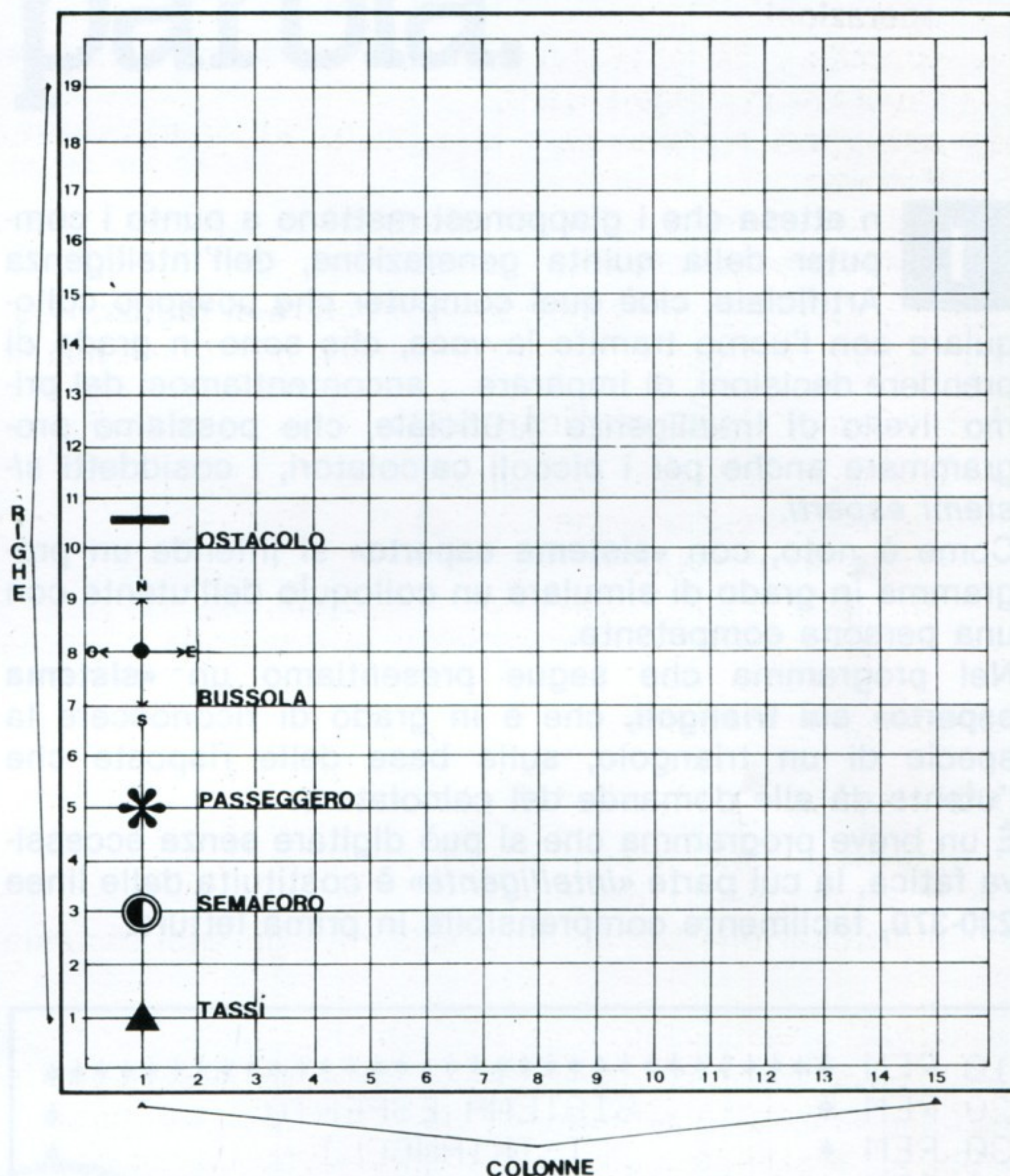
Fig. 4

Il taxi può spegnersi se per esempio vogliamo prendere a bordo un passeggero che non è all'incrocio previsto, oppure lasciarlo quando non è ancora salito sulla macchina.

[illegible]

Con il prossimo numero di LIST si inseriranno degli ostacoli e si introdurranno altre parole che il tassista potrà usare.

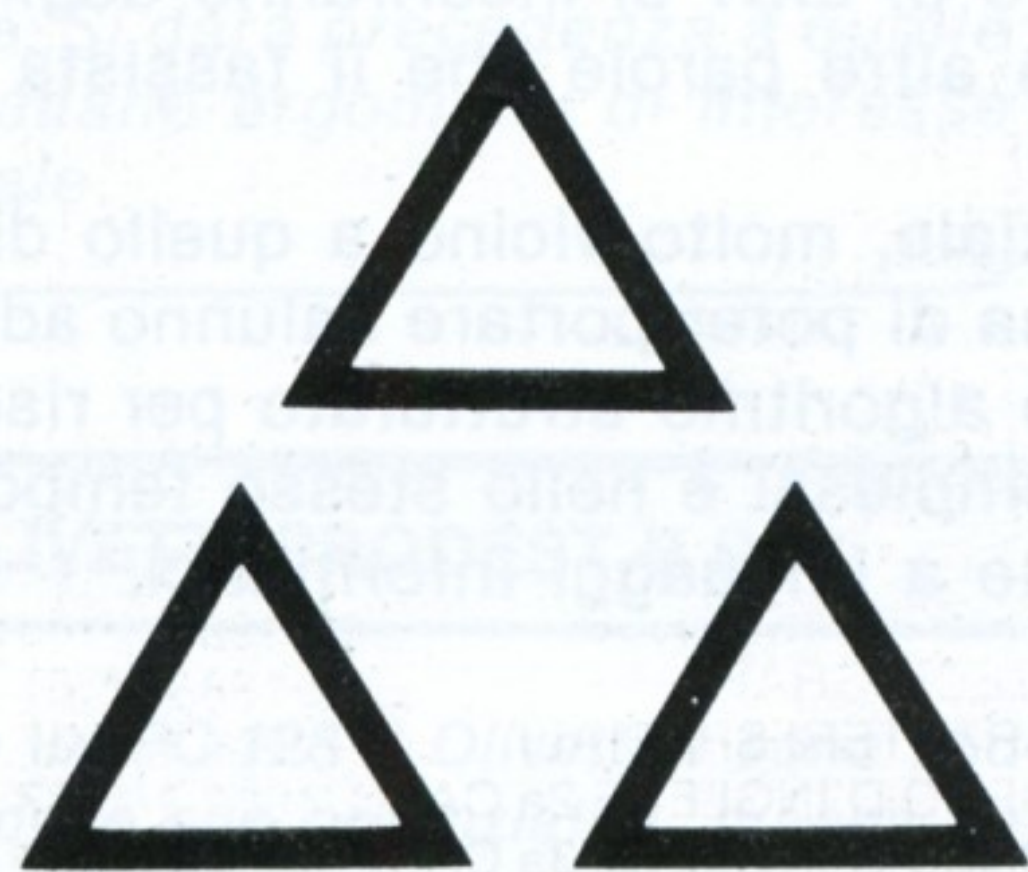
Fig. 3



Abbonamento
“SPECIALE SCUOLA”

Prezzo particolare
per Presidi, Direttori, Insegnanti, Studenti
L. 38.500 (anziché 55.000)

**Il versamento va effettuato sul
c/c post. N. 72609001 indirizzato a
LIST programmi per il tuo Home
Computer - Via F. Stilicone, 111 - 00175
Roma-apponendo nello spazio
per la causale il timbro
della Scuola di appartenenza.**



I TRIANGOLI

*di Giuseppe Festa
per Commodore C64*

In attesa che i giapponesi mettano a punto i computer della quinta generazione, dell'Intelligenza Artificiale, cioè quei computer che possono colloquiare con l'uomo tramite la voce, che sono in grado di prendere decisioni, di imparare..., accontentiamoci del primo livello di Intelligenza Artificiale, che possiamo programmare anche per i piccoli calcolatori, i cosiddetti **sistemi esperti**.

Come è noto, con «**sistema esperto**» si intende un programma in grado di simulare un colloquio dell'utente con una persona competente.

Nel programma che segue presentiamo un «**sistema esperto**» sui triangoli, che è in grado di riconoscere la specie di un triangolo, sulla base delle risposte che l'utente dà alle domande del calcolatore.

È un breve programma che si può digitare senza eccessiva fatica, la cui parte «*intelligente*» è costituita dalle linee 230-370, facilmente comprensibile in prima lettura.

```

10 REM *****
20 REM *          SISTEMA ESPERTO:          *
30 REM *          I TRIANGOLI              *
40 REM *****
50 REM *          DI GIUSEPPE FESTA - ROMA   *
60 REM *****
70 POKE53280,12:POKE53281,11:PRINTCHR$(1
42)
80 PRINT"##### SISTEMA
ESPERTO"
90 PRINT"##### TRIANGOLI"
100 GOSUB2030
110 PRINT"##### SE MI COMUNICHI
LE CARATTERISTICHE"
120 PRINT"##### DEL TRIANGOLO, TI DIRO' DI
CHE COSA"
130 PRINT"##### SI TRATTA."
140 GOSUB2030
200 REM *****
210 REM *          TRIANGOLI              *
220 REM *****
230 INPUT"##### HA TUTTI I LATI UGUALI";C$
240 IFC$="SI"THENPRINT"##### UN TRIANGOLO
EQUILATERO ED EQUIANGOLO":GOSUB900:GOTO
200

```

```

250 INPUT"##### HA DUE LATI UGUALI";D$
260 IFD$="SI"THENPRINT"##### UN TRIANGOLO
ISOSCELE.":GOTO330
270 PRINT"##### UN TRIANGOLO SCALENO"
280 INPUT"##### HA UN ANGOLO RETTO";E$
290 IFE$="SI"THENPRINT"##### UN TRIANGOLO
RETTANGOLO SCALENO.":GOSUB900:GOTO200
300 INPUT"##### HA UN ANGOLO OTTUSO";O$
310 IFO$="SI"THENPRINT"##### UN TRIANGOLO
SCALENO OTTUSANGOLO.":GOSUB900:GOTO200
320 IFO$="NO"THENPRINT"##### UN TRIANGOLO
SCALENO ACUTANGOLO.":GOSUB900:GOTO200
330 INPUT"##### HA UN ANGOLO RETTO";F$
340 IFF$="SI"THENPRINT"##### UN TRIANGOLO
RETTANGOLO ISOSCELE.":GOSUB900:GOTO200
350 INPUT"##### HA UN ANGOLO OTTUSO";G$
360 IFG$="SI"THENPRINT"##### UN TRIANGOLO
ISOSCELE OTTUSANGOLO.":GOSUB900:GOTO200
370 PRINT"##### UN TRIANGOLO ISOSCELE ACU
TANGOLO.":GOSUB900:GOTO200
900 REM *****
910 REM *          SUBROUTINES          *
920 REM *****
1000 REM *****
1010 REM *          CONTINUI?          *
1020 REM *****
1030 PRINT"##### CONTINUI? (S/N)
1040 GETX$:IFX$=""THEN1040
1050 IFX$="S"THENRETURN
1060 END
2000 REM *****
2010 REM *          PREMI UN TASTO          *
2020 REM *****
2030 PRINTTAB(25)"##### PREMI UN TASTO"
2040 GETA$:IFA$=""THEN2040
2050 RETURN

```


I PROGRAMMI DI LIST

PER RICHIESTE VEDI
"SERVIZIO PROGRAMMI"

COMMODORE 128

LIST N

BUSINESS GRAPHICS	11/1986
TUNDER BALL	11/1986
AR.CON	12/1986
TOP-GUN	1/1987
Progressioni ARITMETICHE	2/1987
ROMPICAPO (puzzle)	2/1987
TECNO	3/1987
RAINBOW	5/1987
EASY DATA	5/1987

COMMODORE 64/128

LIST N

DESIGN LAB	10-11/1985
TUTTO IL CALCIO...	12/1985
RUBRICA TEL.	1-2/1986
VIDEOSINT	1-2/1986
RADICE QUADRATA	3/1986
ASTON 64 TITOLATRICE	3/1986
CATALOGO PROGRAMMI	4/1986
AFFINITÀ DI COPPIA	4/1986
TEST DI GEOMETRIA	5-6/1986
SUPER UTILITY	5-6/1986
ISTOGRAMMI 3-D	5-6/1986
NOTE DIRECTORY	9/1986
WINDOW 64	10/1986
SPRITE BUSTER	11/1986
CLASSIFICA CAMP. A-B	11/1986
DISEGNO E STAMPA	12/1986
CONVERSIONE NUMERI	12/1986
SOGLIANDO IL DIZIONARIO	1/1987
FORMULA 3000	1/1987
EQUIVALENZE	1/1987
LE TABELLINE	2/1987
CALCOLATRICE SCIENTIFICA	2/1987
DISCUSSIONE DI FUNZIONE	2/1987
IN GIRO PER L'EUROPA	3/1987
AMMORTAMENTO INTERESSI	3/1987
COPERTINA PER CASSETTE	5/1987

COMMODORE 16/PLUS 4

LIST N

CHIMICA	10-11/1985
ASTON 16	12/1985
CHAR 16 GEN. CARAT.	3/1986
CAMPIONATO DI CALCIO	4/1986
BIORITMI	7-8/1986

SEGA SC 300

LIST N

MATH SOFTWARE	8-9/1985
GEOMETRIA 1	10-11/1985
GRAPH 3000	1-2/1986
TOPOGRAPH 3000	3/1986
DRAWER	4/1986
BAKER	5-6/1986
HANOI	7-8/1986
GFORZA 4	11/1986
OBLIN	9/1986
DERBY	10/1986
R.A.F.	10/1986
TOP-SECRET	12/1986
POKER	12/1986

SHARP MZ 700

LIST N

ELENCO FORNITORI	10-11/1985
SETTE E MEZZO	10-11/1985
LO STRISCIONE	12/1985
ELENCO CLIENTI	1-2/1986
IL MURO	1-2/1986
SHARPOORGANO	1-2/1986
THE STING	3/1986
CACCIA ALLE LETTERE	1-2/1986
GESTIONE MAGAZZINO	4/1986
FILETTO	5-6/1986
BUSINESS MANN	5-6/1986
CONTABILITÀ DOMESTICA	7-7/1986
GRAFICI AZIENDALI	9/1986
TOTOCALCIO	10/1986
SCOPRI LA CARTA	10/1986
BATTAGLIA NAVALE	11/1986
CONOSCERE L'ANATOMIA	12/1986
ESAME DI GEOMETRIA	12/1986
ENALOTTO	1/1987
CONOSCERE L'ANATOMIA	1/1987
FUNZIONI 3D	2/1987
COPERTINE PER CAS.	2/1987
CODICE COLORE RESISTENZE	2/1987
WORD PROCESSOR	3/1987
CODICE FISCALE	5/1987
COORDINATE CARTESIANE	5/1987

TEXAS TI 99/4A

LIST N

SCHEDA P. MONDRIAN	8-9/1985
LA COMETA DI HALLEY	10-11/1985
LEONARDO	12/1985
THE LORD OF SPIDERS	12/1985
MAYA	1-2/1986
DECODIFICATORE	1-2/1986
CONV. ESADEC.-DEC.	3/1986
BOWLING	3/1986
PROGETTO D.A.C.I.A.	4/1986
GALAXI ADVENTURE	4/1986
SIMON TEXAS	5-6/1986
SKIING IN THE NIGHT	5-6/1986
DISEGNO IN HI-RES	7-8/1986
VIDEO PRINT	11/1986
MERRY GO ROUND	11/1986
CHERNOBYL	9/1986
PASSAGGIO SU GIZA	9/1986
MARS WARS	10/1986
VIDEOCONCERTO	12/1986
INTERCEPTOR	12/1986
CALENDARIO PERPETUO	1/1987
GRAZY BALL	2/1987
GOLF	3/1987

SISTEMI MSX

LIST N

POKER	8-9/1985
GALACTICA	10-11/1985
PAINT WITH DRAW	12/1985
NUDO ARTISTICO	5/1987
EDIT SPRITE	5/1987
GRAFICA MATEMATICA	5/1987

DATA WRITER	1-2/1986
ROAD RACE	1-2/1986
GHOST-MAZE	3/1986
ROMPICAPO	1-2/1986
WIMBLEDON	4/1986
SUPER SORT	5-6/1986
THE DIES	7-8/1986
MASTERMIND MSX	9/1986
ORD. ALFABETICO	10/1986
C/C BANCARIO	10/1986
BONUS MALUS	11/1986
TORNIO ELETTRONICO	11/1986
CHE GIORNO ERA	12/1986
CORSO DI INGLESE	12/1986
IL TECNIGRAFO	1/1987
CARATTERI SPECIALI	1/1987
CORSO D'INGLESE 2a CAS.	1/1987
CORSO D'INGLESE 3a CAS.	2/1987
SCARABEUS	2/1987
TITOLATRICE A 20 INGRAND.	3/1987
L'INVENTA SPRITE	3/1987
CHI È	3/1987
CORSO INGLESE 4 ^a CASSETTA	3/1987

ZX SPECTRUM

LIST N

FUSO ORARIO	1-2/1986
CAMP 1	1-2/1986
SLALOM	3/1986
TOT 13	3/1986
DATA BASE	4/1986
RENUMBER	4/1986
ROTOR	5-6/1986
PAINT	5-6/1986
TIC-TAC-TOE 3D	7-8/1986
Q.I.	9/1986
IL TEST DELL'AMORE	10/1986
GEOMETRIA PIANA	10/1986
Y 30	11/1986
PIANO BAR	11/1986
ROULETTE	12/1986
DOUBLE	12/1986
AIRAM	1/1987
TESEO	1/1987
DAMA	2/1987
MERIDIANA	5/1987
STARLITH	5/1987
ECLISLUNA	3/1987

OLIVETTI PRODEST PC128

LIST N

QUARKSTYLE	1/1987
MASTERMIND	1/1987
ALFABETO PAZZO	2/1987
SEMPLIFICAZIONE DI FRAZIONI	3/1987
OTHELLO	4/1987
MEMORY	4/1987
IL VIAGGIOSCOPIO	5/1987

OLIVETTI PRODEST PC128S

LIST N

WINDOW EDITOR	2/1987
SPHEROGRAPH	3/1987
OLICALC	4/1987
CONTABILITÀ DOMESTICA	5/1987



SERVIZIO PROGRAMMI

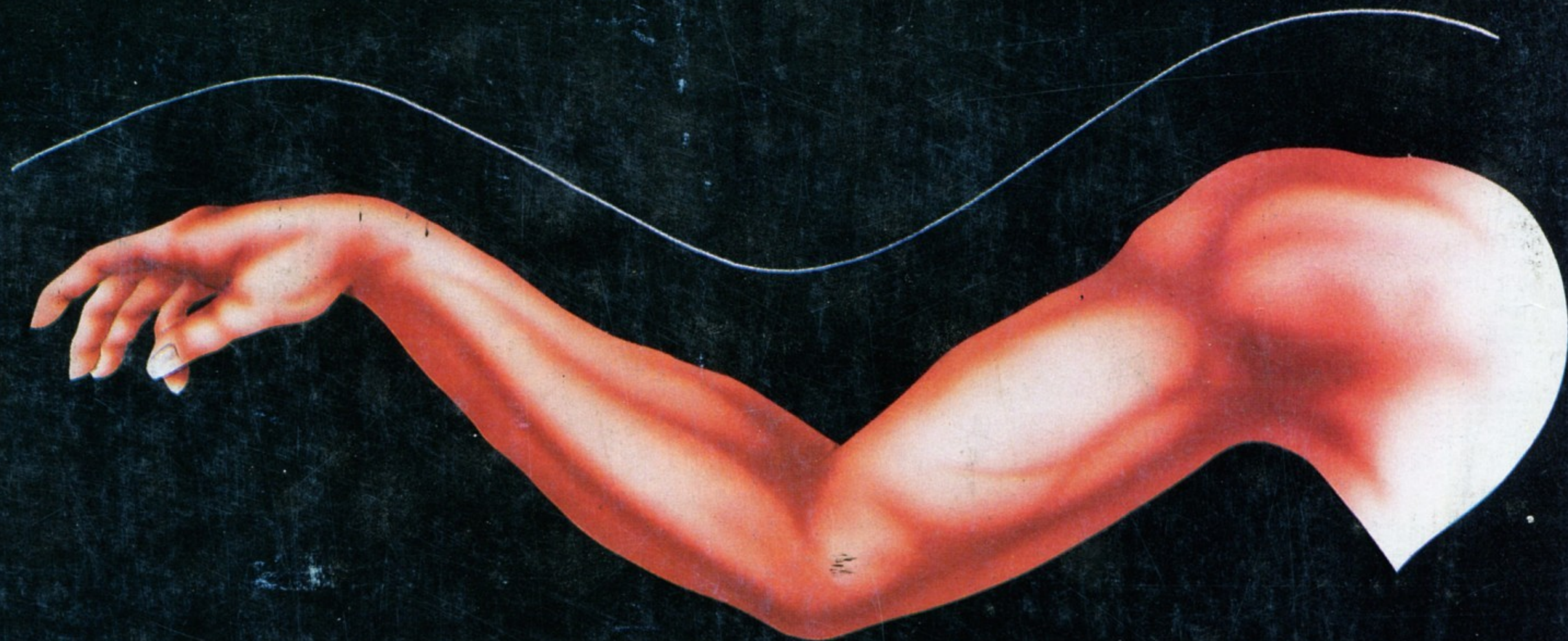
I programmi, pubblicati in questo e nei precedenti numeri di **LIST**, sono disponibili in cassetta o (floppy disk solo per Commodore) I lettori possono richiedere uno o più programmi, specificandone i titoli, e inviando, quale contributo spese, per ogni programma:

Lire 10.000 per la cassetta, Lire 12.000 per il floppy disk (Commodore) e L. 15.000 per Olivetti PC 128S

Tale contributo è comprensivo delle spese di imballo e spedizione a domicilio.

La richiesta deve essere fatta a mezzo vaglia postale; indirizzato a EDICOMP s.r.l. Via F. Stilicone, 111 - 00175 Roma oppure con assegno bancario N.T. in busta chiusa.

Biometal Robot... tecnologia del «2000»



Robot 5 assi azionato da Biometal.

DH101 non usa pistoni o motori: Biometal è una speciale lega capace di modificare la propria forma se percorsa da corrente.

Biometal in forma di filo (0.15 mm) aziona **DH101 Robot** come un muscolo umano.

Caratteristiche: — 5 assi open loop.

— modi di controllo: manuale, trace, E-Z link (seriale), parallelo Centronics, demo1, demo2.

— Voi stessi potete registrarlo nei movimenti.

Esperienze con Biometal: — DH620 starting kit.

— Biometal guide book.

Biometal è un marchio registrato della **TOKI Corporation**. **DH101 Robot** è coperto da brevetto.



RAVASI ROBOTICS è una divisione della RAVASI I.M.S.

RAVASI ROBOTICS

RAVASI ROBOTICS è una divisione della RAVASI I.M.S.
s.n.c. di Ravasi Felice e Fabrizio & C.

Via Cartiera, 1 - 22050 Brivio (Co) Italia

Telefono 039/53.21.433

Telex 380161 RAVASI LCCI

Cognome _____

Nome _____

Indirizzo _____

CAP _____ Città _____ Prov. _____

Sono interessato a: ☐ DH 101 Robot
☐ DH 620 starting kit
☐ Biometal guide book

Il mio personal computer è _____