

# PERSONAL SOFTWARE



ANNO 4 N. 30  
LUGLIO/AGOSTO 1985  
L. 4.000

LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER



**GEO-RACE  
CON LO SPECTRUM**

**EASY VIDEO PER C 64**

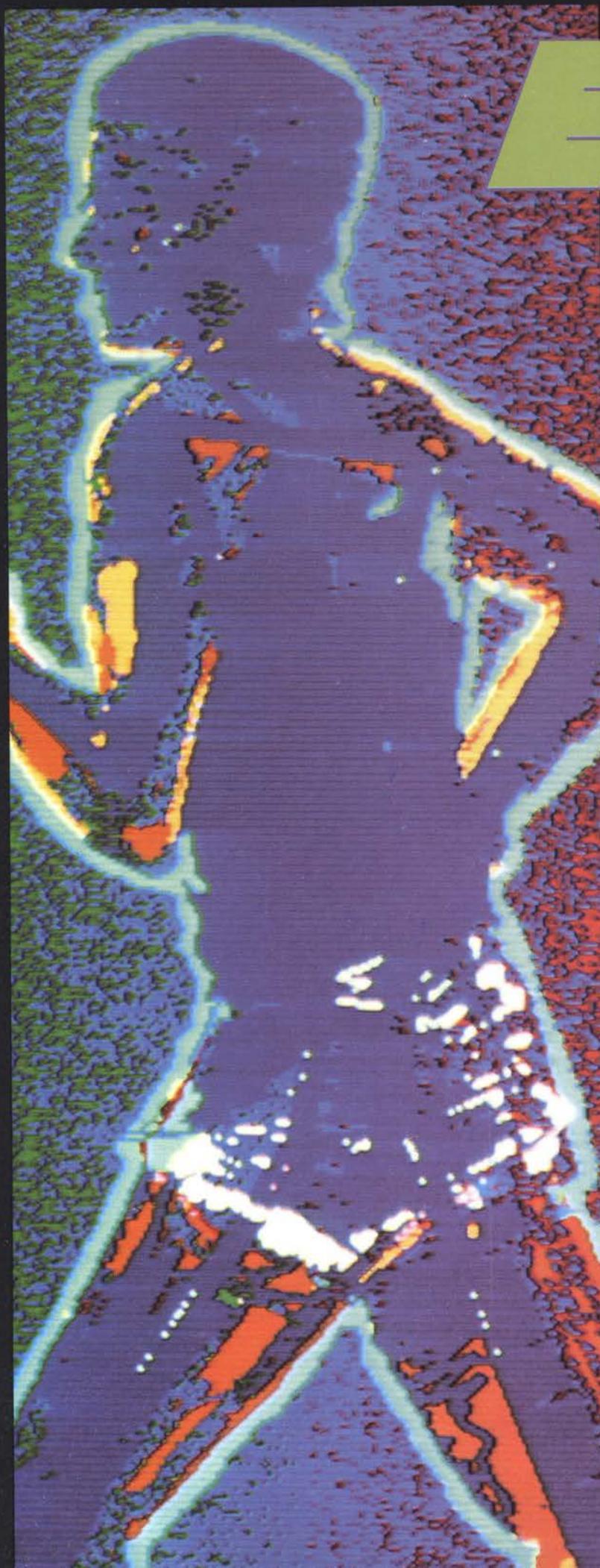
**HI-RES PER TI99/4A**

**DATA BASE PER C 16**

**STATISTICA A PIÙ  
DIMENSIONI CON L'APPLE**

**DISEGNATORE:  
UN PROGRAMMA PER MSX**

# NON FARTI SUPERARE DAL PROGRESSO

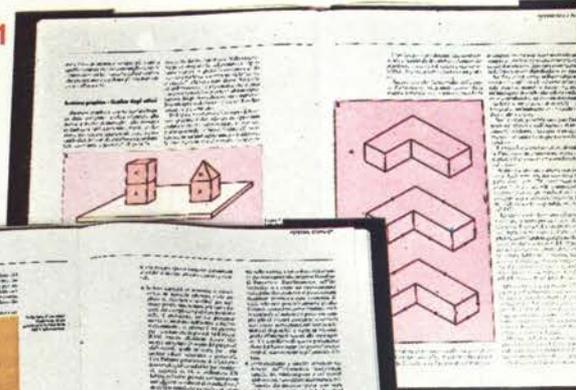


## Aggiornati con gli "Aggiornamenti"

### ENCICLOPEDIA DI ELETTRONICA & INFORMATICA

**20** FASCICOLI SETTIMANALI DA RILEGARE  
IN DUE NUOVI E SPLENDIDI VOLUMI

1



2

**PER TE, PER IL TUO LAVORO, PER I TUOI STUDI...**

#### 1 **Aggiornamenti**

le nuove conquiste dell'Elettronica di Base, delle Comunicazioni, dell'Elettronica Digitale, dei Microprocessori, dell'Informatica...

#### 2 **Il personal computer**

tutto quello che c'è da sapere sul Personal Computer: che cos'è e cosa fa; come fa e come si fa; i linguaggi di programmazione; le applicazioni...

**E.I. si aggiorna e ti aggiorna**

**il 18 settembre**

batti sul tempo il progresso!  
Corri in edicola a comprare il primo fascicolo.

**Solo L. 2500**



**ANCORA UNA VOLTA  
"PRIMI SUL FUTURO"**



**GRUPPO  
EDITORIALE  
JACKSON**

SAN FRANCISCO - LONDRA - MILANO



## **Smau: il giro del mondo in 91.000 metri quadrati**

Smau: chi lo visita farà un entusiasmante giro del mondo in 91.000 mq.

Qui infatti troverà tutte le novità dei più importanti produttori mondiali.

Qui troverà esperti capaci di consigliare le soluzioni più aderenti al futuro dell'azienda e dell'organizzazione del lavoro.

Troverà la 18ª edizione del Premio Smau Industrial Design; troverà Convegni e Seminari; troverà lo Spazio Giovani. Troverà il mondo intero: tutto racchiuso in 91.000 metri quadrati.



22° Salone Internazionale per l'Ufficio: sistemi per l'informatica, la telematica, le comunicazioni, macchine, arredamento per l'ufficio

ENTE GESTIONE MOSTRE COMUFFICIO

Quartiere Fiera Milano  
19-24 Settembre 1985

Contemporaneamente, 3ª EIMU,  
Esposizione Internazionale Mobili Ufficio

# Novità firmate Jackson.



Giulio Carducci

## LE APPLICAZIONI DEL COMPUTER NELL'UFFICIO MODERNO

Questo libro vuole essere un'introduzione all'informatica e, nel contempo, una guida all'utilizzo consapevole dello strumento del giorno, il personal computer, nell'ufficio moderno e nello studio professionale. È rivolto pertanto, ad un vasto pubblico: addetti ai vari settori dell'azienda, quadri, dirigenti, ingegneri, architetti, professionisti in genere.

Cod. 407H Pag. 132 Lire 23.000

Michael Browne

## UNITÀ A DISCHI PER MICROCOMPUTER

Il libro, destinato a lettori con una buona conoscenza di base dell'uso di un calcolatore e del linguaggio BASIC, descrive il funzionamento dell'unità a dischi di un personal computer e il significato dei comandi relativi, con particolare attenzione per le diverse tecniche di gestione dei file su disco.

Il libro è arricchito di diversi programmi esempio in BASIC Commodore 4.0, relativi alle diverse tecniche di organizzazione e gestione dei file.

Cod. 300P Pag. 156 Lire 15.000

Mauro Salvemini

## URBANISTICA E INFORMATICA

Sempre più vaste sono le applicazioni dell'informatica in ogni settore della vita e del lavoro, e in particolare quelle della computer grafica. Neanche mestieri e professioni con una origine antica come l'architetto o l'urbanistica si "salvano" dall'onda dell'informatica.

Un entusiasta utilizzatore di quest'area culturale racconta in questo libro per i colleghi e gli studenti di architettura e di urbanistica tutti i vantaggi e gli avanzamenti possibili nella progettazione e lo studio di nuovi edifici, complessi edilizi o urbani grazie alla computer grafica.

Cod. 801P Pag. 224 Lire 30.000

Graziella Tongoni

## LA COMUNICAZIONE CAMBIATA

Questo libro vuole offrire una occasione a tutti i suoi lettori per analizzare, scoprire e potenziare le proprie capacità comunicative e divenire protagonisti reali di una ampia trasformazione in corso, di cui uno degli aspetti più evidenti è costituito dall'espandersi progressivo delle tecnologie.

Cod. 538P Pag. 112 Lire 10.500

Arthur Naiman

## WORD STAR

Perché un computer, anche di categoria micro, sia utile nell'attività di ufficio è essenziale che permetta anche di effettuare l'elaborazione dei testi, con tutte le funzioni che ciò comporta. I prodotti software a ciò destinati sono oggi moltissimi ma solo alcuni emergono per completezza e flessibilità: fra questi Wordstar è sicuramente uno dei più riusciti.

Cod. 525P Pag. 222 Lire 23.000



GRUPPO EDITORIALE JACKSON

## La biblioteca che fa testo.

ritagliare (o fotocopiare) e spedire in busta chiusa a:  
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

### CEDOLA DI COMMISSIONE LIBRARIA

#### VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

#### Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca  Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

n° \_\_\_\_\_  Allego fotocopia di versamento su vaglia postale a voi intestato

Nome \_\_\_\_\_  
Cognome \_\_\_\_\_  
Via \_\_\_\_\_  
Cap \_\_\_\_\_ Città \_\_\_\_\_ Prov. \_\_\_\_\_  
Data \_\_\_\_\_ Firma \_\_\_\_\_

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. \_\_\_\_\_

ORDINE MINIMO L. 50.000

# SOMMARIO

<b>LA SIMULAZIONE DINAMICA DI FENOMENI CONTINUI</b> <i>di Franco Sardo</i>	COMMODORE 64	<b>16</b>
<b>AGENDA TELEFONICA</b> <i>di Fabio Fanecco</i>	CASIO PB-100-200-300	<b>24</b>
<b>DATA BASE</b> <i>di Claudio Poma</i>	COMMODORE 16	<b>27</b>
<b>EASY VIDEO 64</b> <i>di Maurizio Paolinelli</i>	COMMODORE 64	<b>41</b>
<b>CUBO MAGICO</b> <i>di Compagiani</i>	ZX SPECTRUM	<b>47</b>
<b>HI-RES</b> <i>di Sergio Borsani</i>	TEXAS TI99/4A	<b>52</b>
<b>STATISTICA A PIU' DIMENSIONI</b> <i>di Roberto Brunialti</i>	APPLE II	<b>61</b>
<b>GEO-RACE</b> <i>di Federico Lo Cicero</i>	ZX SPECTRUM	<b>77</b>
<b>ROUTINE 2 PER SPECTRUM</b> <i>di Ivano Parbuono</i>	ZX SPECTRUM	<b>81</b>
<b>DISEGNATORE PER MSX</b> <i>di Paolo Ferrami</i>	MSX	<b>86</b>

## RUBRICHE

<b>EDITORIALE</b> <i>di Riccardo Paolillo</i>		<b>7</b>
<b>POSTA</b>		<b>9</b>
<b>PERSONAL NEWS</b> <i>a cura di Marco Giacobazzi</i>		<b>12</b>
<b>I SEGRETI DEI PERSONAL:</b>		
<b>Uso delle label simboliche</b> <i>di Lino Squarza</i>	COMMODORE 64	<b>96</b>
<b>Caratteri &amp; colori</b> <i>di Martino Sangiorgio</i>	SHARP MZ-700	<b>102</b>
<b>Da cassetta a microdrive</b> <i>di Marcello Spero</i>	ZX SPECTRUM	<b>104</b>
<b>Quali segreti ci riserva la stampa?</b> <i>di Sergio Borsani</i>	TEXAS TI99/4A	<b>107</b>
<b>PERSONAL MARKET</b>		<b>110</b>



Questo mese: Disegnatore per MSX, come scoprirsi valenti pittori con questo splendido programma in standard MSX.

**ANNO 4**  
**N. 30**  
**LUGLIO-AGOSTO 1985**

**CON INSERTO SUPERBIT**  
64 PAGINE DI SOFTWARE PER IL TUO PERSONAL

# è in edicola il nuovo numero

*Bit, la prima  
e più diffusa rivista  
di personal computer  
e accessori*



**UNA  
PUBBLICAZIONE  
DEL  
GRUPPO  
EDITORIALE  
JACKSON**

SAN FRANCISCO-LONDRA-MILANO



**B**itest  
Atari 130 XE

**S**lalom  
parallelo:  
HP PC Integrato  
contro  
GRiD Compass II

**E**nterprise  
Sixty-Four

**C**orso  
di Assembly  
per C 64

**O**ver BASIC  
per Spectrum

**C**ontabilità  
casalinga  
con il C 64

**Q**L Art

**D**isegnare  
sullo schermo  
con un MSX

**SPECIALE:  
DATA-BASE**

## Musica, maestro!

**È** sicuramente vero che la musica riveste un ruolo molto importante per la maggior parte delle persone. La grande varietà di generi, classici o moderni, garantisce a chiunque di soddisfare i propri gusti. Ognuno ascolta la musica con uno stato d'animo differente: chi nei momenti di tristezza, chi in quelli di gioia.

Ma la musica serve anche a distrarsi, sognare, meditare, rilassarsi, addormentarsi o lavorare: è, in definitiva, una componente fondamentale della nostra vita.

Inevitabile quindi che la ricerca di nuove e sempre più sofisticate funzioni di cui dotare i personal computer si orientasse decisamente in questo senso.

L'inizio di musica con il calcolatore, a dir la verità, era stato un po' deludente anche per gli audiofili meno esigenti. Ancora pochi anni fa il massimo che si riusciva ad ottenere era la possibilità di modificare la durata di un "beep" monofonico dando così vita a melodie di struggente tristezza.

La strada era però segnata e in men che non si dica vennero lanciati sul mercato home computer a basso costo, ma con prestazioni sonore strepitose: addirittura tre voci, indipendenti e programmabili, garantite da un apposito chip dedicato. È proprio il caso di dire che la musica cambiò; non più suoni gracchianti e cacofonici, ma motivi di piacevole ascolto.

Il fatto nuovo e importante, però, non era semplicemente quello di far suonare un pezzo al proprio calcolatore. La possibilità di poter mettere le mani su uno "strumento" musicale dall'uso molto più semplificato rispetto a quelli tradizionali, ha sicuramente contribuito ad avvicinare maggiormente molte persone alla musica.

Grazie a semplici programmi di sintesi musicale, o addirittura veri e propri corsi di facile comprensione come *7 Note Bit* del Gruppo Editoriale Jackson in collaborazione con la SIEL, chiunque può improvvisarsi musicista e provare l'emozione di suonare il proprio pezzo preferito.

La cosa comunque non finisce qui: anche i veri musicisti si stanno accorgendo di questi nuovi mezzi che la tecnologia propone. Sono sempre più numerosi i personal che spuntano tra tastiere e mixer sui palcoscenici o nelle sale di incisione. A volte sono utilizzati per programmare l'esecuzione di altri strumenti, ma spesso anche per suonare, soprattutto parti ritmiche.

Se utilizzati intelligentemente, come spesso accade, non tolgono nulla alla creatività dell'artista, ma anzi ne mettono in risalto la sensibilità.

# Debug

L'articolo Bianchi & Neri per Apple pubblicato nel numero 28 - Maggio 1985 - è opera di Franco e Antonio Massaro e non di Stefano Guarinelli come erroneamente riportato. Ce ne scusiamo con gli interessati.

Inoltre, per esigenze tipografiche, non è stato pubblicato il listato Assembly della routine contrariamente a quanto scritto nell'articolo.

Lo pubblichiamo ora in quanto può essere molto utile a chi intendesse studiare a fondo il programma per convertirlo a un altro personal.

```

----- NEXT OBJECT FILE NAME IS SUBROUTINE.OBJD
6001:      1      ORG $6001
6001:      2 *
6001:      3 *ASSEGNA ALLE LO-
6001:      4 *CAZIONI DI MEMO-
6001:      5 *RIA LE RELATIVE
6001:      6 *LABELS.
6001:      7 *
0300:      8 NUME EQU $300
0301:      9 CORSO1 EQU $301
0302:     10 CORSO2 EQU $302
0303:     11 CORSO3 EQU $303
0304:     12 CORSO4 EQU $304
0305:     13 NUMER EQU $305
0327:     14 PRENER EQU $327
0328:     15 PREBIA EQU $328
034B:     16 DEPOS1 EQU $34B
034C:     17 DEPOS2 EQU $34C
034D:     18 DEPOS3 EQU $34D
034E:     19 NERI EQU $34E
034F:     20 BIANCHI EQU $34F
0350:     21 CONT EQU $350
6001:     22 *
  
```

```

6001:      23 *INIZIO SUBROUT.
6001:      24 *CREA IL NUMERO
6001:      25 *IN CORSO.
6001:      26 *
6001:A9 00      27      LDA $#00
6003:8D 01 03  28      STA CORSO1
6006:EE 01 03  29 VAR1  INC CORSO1
6009:AD 01 03  30      LDA CORSO1
600C:C9 0A     31      CMP $#0A
600E:30 01     32      BMI PROSEG
6010:60        33      RTS
6011:A9 00     34 PROSEG LDA $#00
6013:8D 02 03  35      STA CORSO2
6016:EE 02 03  36 VAR2  INC CORSO2
6019:AD 02 03  37      LDA CORSO2
601C:C9 0A     38      CMP $#0A
601E:FO E6     39      BEQ VAR1
6020:A9 00     40      LDA $#00
6022:8D 03 03  41      STA CORSO3
6025:EE 03 03  42 VAR3  INC CORSO3
6028:AD 03 03  43      LDA CORSO3
602B:C9 0A     44      CMP $#0A
602D:FO E7     45      BEQ VAR2
  
```

```

602F:A9 00      46      LDA $#00
6031:8D 04 03  47      STA CORSO4
6034:EE 04 03  48 VAR4  INC CORSO4
6037:AD 04 03  49      LDA CORSO4
603A:C9 0A     50      CMP $#0A
603C:FO E7     51      BEQ VAR3
603E:A2 00     52      LDX $#00
6040:AD 04 03  53      LDA CORSO4
6043:DD 01 03  54 CONFR  CMP CORSO1,X
6046:FO EC     55      BEQ VAR4
6048:EB        56      INX
6049:EO 03     57      CPX $#03
604B:DD F6     58      BNE CONFR
604D:AD 03 03  59      LDA CORSO3
6050:CD 01 03  60      CMP CORSO1
6053:FO D0     61      BEQ VAR3
6055:CD 02 03  62      CMP CORSO2
6058:FO CB     63      BEQ VAR3
605A:AD 02 03  64      LDA CORSO2
605D:CD 01 03  65      CMP CORSO1
6060:FO B4     66      BEQ VAR2
6062:          67 *
6062:          68 *CONFRONTA IL NU-
6062:          69 *MERO IN CORSO
6062:          70 *CON UN NUMERO
6062:          71 *GIOCATO DAL
6062:          72 *COMPUTER E CAL-
6062:          73 *COLA IL PUNTEG-
6062:          74 *GIO OTTENUTO
6062:          75 *DAL CONFRONTO.
6062:          76 *
6062:A9 04      77      LDA $#04
6064:8D 48 03  78      STA DEPOS1
6067:AD 00      79      LDY $#00
6069:8C 50 03  80      STY CONT
606C:EE 50 03  81 CONTINUA INC CONT
606F:8C 4F 03  82      STY BIANCHI
6072:8C 4E 03  83      STY NERI
6075:98        84      TYA
6076:AE 50 03  85      LDX CONT
6079:CA        86 LOOP  DEX
607A:FD 07     87      BEQ EXIT
607C:19        88      CLC
607D:6D 48 03  89      ADC DEPOS1
6080:4A 79 60  90      JMP LOOP
6083:AC        91 EXIT  TAX
6084:8E 4C 03  92      STX DEPOS2
6087:8C 4E 03  93 CONFR LDA NUMER,X
608A:D9 01 03  94      CMP CORSO1,X
608D:DD 03     95      BNE INCRY
608F:2D C7 60  96      JSR CALCBN
6092:C8        97 INCRY  INY
6093:C0 04     98      CPY $#04
6095:DD F0     99      BNE CONF
6097:AD 00     100     LDY $#00
6099:EB       101     INX
609A:8A       102     TXA
609B:38       103     SEC
609C:ED 4C 03 104     SBC DEPOS2
609F:C9 0A    105     CMP $#04
60A1:DD E4    106     BNE CONF
60A3:         107 *
60A3:         108 *CONFRONTA IL PUN-
60A3:         109 *TEGGIO DEL NUME-
60A3:         110 *RO IN CORSO
60A3:         111 *CON QUELLO DI
60A3:         112 *GIOCO.
60A3:         113 *
60A3:AD 50 03 114     LDA CONT
60A6:18       115     CLC
60A7:6D 50 03 116     ADC CONT
60AA:AA       117     TAX
60AB:BD 27 03 118     LDA PRENER,X
60AE:CD 4E 03 119     CMP NERI
60B1:DD 81     120     BNE VAR4
60B3:BD 28 03 121     LDA PREBIA,X
60B6:CD 4F 03 122     CMP BIANCHI
60B9:DD 09     123     BNE RINVIA
60BB:AD 50 03 124     LDA CONT
60BE:CD 00 03 125     CMP NUME
60C1:DD A9     126     BNE CONTINUA
60C3:60       127     RTS
60C4:4C 34 60 128 RINVIA JMP VAR4
60C7:         129 *
60C7:         130 *SUBROUT. CALCBN.
60C7:         131 *
60C7:8A       132 CALCBN TXA
60CB:38       133     SEC
60C9:ED 4C 03 134     SBC DEPOS2
60CC:8D 4D 03 135     STA DEPOS3
60CF:CC 4D 03 136     CPY DEPOS3
60D2:F0 04     137     BEQ NER
60D4:EE 4F 03 138     INC BIANCHI
60D7:60       139     RTS
60DB:EE 4E 03 140 NER  INC NERI
60DB:60       141     RTS
  
```

034F BIANCHI	60C7 CALCBN	6043 CONFR	6087 CONF
0350 CONT	606C CONTINUA	0301 CORSO1	0302 CORSO2
0303 CORSO3	0304 CORSO4	034B DEPOS1	034C DEPOS2
034D DEPOS3	60B3 EXIT	6092 INCRY	6079 LOOP
60DB NER	034E NERI	0300 NUME	0305 NUMER
0328 PREBIA	0327 PRENER	6011 PROSEG	60C4 RINVIA
6006 VAR1	6016 VAR2	6025 VAR3	6034 VAR4
0300 NUME	0301 CORSO1	0302 CORSO2	0303 CORSO3
0304 CORSO4	0305 NUMER	0327 PRENER	0328 PREBIA
034B DEPOS1	034C DEPOS2	034D DEPOS3	034E NERI
034F BIANCHI	0350 CONT	6006 VAR1	6011 PROSEG
6016 VAR2	6025 VAR3	6034 VAR4	6043 CONFR
606C CONTINUA	6079 LOOP	60B3 EXIT	6087 CONF
6092 INCRY	60C4 RINVIA	60C7 CALCBN	60DB NER

## Compilatori ed interpreti

Posseggo uno ZX Spectrum da poco tempo e sto cominciando a realizzare i miei primi programmi. Utilizzo il linguaggio BASIC di cui è dotato il mio personal. Ho letto sul manuale d'uso che tale linguaggio è interpretato, mentre anche su altre riviste ho letto che esistono linguaggi compilati. Gradirei sapere in cosa differiscono, se posso utilizzare un compilatore con il mio calcolatore e che benefici ne ricaverai.

*Stefano Bennini  
Pistoia*

Cominciamo dall'inizio. Compilatori ed interpreti sono essi stessi dei particolari programmi che hanno il compito di trasformare una serie di istruzioni scritte in un linguaggio di programmazione ad alto livello in codici macchina direttamente eseguibili dal microprocessore.

I linguaggi ad alto livello sono quelli normalmente usati da chi sviluppa software applicativo e vengono così definiti perché la loro sintassi li rende più vicini al linguaggio naturale (quello parlato) a differenza di quanto succede per i codici macchina, che sono una serie di numeri binari. Se, ad esempio, consideriamo il BASIC, che è senza dubbio il linguaggio ad alto livello universalmente diffuso tra gli utenti di home computer, possiamo osservare che un programma è costituito da una serie di istruzioni (paragonabili a frasi nel linguaggio naturale) formate da parole chiave in lingua inglese legate fra loro mediante particolari regole sintattiche e logiche. In pratica, mediante un programma, diamo al nostro calcolatore delle regole di funzionamento utilizzando un linguaggio comprensibile sia a noi che al processore. Il calcolatore riconosce le istruzioni date e le esegue, grazie proprio a compilatori ed interpreti.

Infatti ogni istruzione scritta in BASIC, ad esempio una semplice Print, deve essere tradotta in una serie di codici direttamente eseguibili dal microprocessore, che appartengono al suo set di istruzioni di base (fra

le 50 e le 100 per i micro più diffusi).

La differenza tra interpreti e compilatori consiste nel modo in cui viene effettuata la traduzione e l'esecuzione del programma. L'interprete esamina una istruzione, la traduce nella sequenza di istruzioni elementari e quindi la esegue. Il tutto viene fatto istruzione per istruzione e quindi la fase di traduzione è contemporanea a quella di esecuzione.

Il compilatore, invece, a partire da un programma in linguaggio ad alto livello ne genera un altro in codice oggetto direttamente eseguibile dal processore. Ogni richiesta di esecuzione farà quindi direttamente riferimento a quest'ultimo codice e non più al programma sorgente.

In termini pratici l'esecuzione di programmi compilati avviene molto più velocemente, in quanto non viene ripetuta ogni volta la fase di traduzione.

D'altra parte è molto più comodo sviluppare programmi utilizzando linguaggi interpretati: infatti dopo ogni modifica il programma è immediatamente eseguibile, mentre con linguaggi compilati occorre ripetere la fase di traduzione nel codice oggetto.

Inoltre i compilatori sono in generale più complessi e dall'uso meno immediato anche se normalmente garantiscono soluzioni più professionali.

Anche per lo Spectrum, come per quasi tutti gli home computer, sono disponibili in commercio compilatori ed interpreti per diversi linguaggi.

Per quanto riguarda in particolare i compilatori ci risulta esistano le versioni per i linguaggi BASIC, Forth, C e Microprolog.

## Software cercasi

Pochi mesi fa acquistai un Commodore Plus 4 convinto dalle sue caratteristiche decisamente interessanti. Ora a distanza di tempo sono un po' deluso, in quanto ho difficoltà a reperire del software per il mio calcolatore. Esiste la possibilità di utilizzare i programmi da voi pubblicati per C 64 direttamente sul mio Plus 4?

*Antonio Maletti  
Roma*

## POSTA



Il C 64 e il Plus 4 sono due macchine piuttosto diverse. Il C 64 è dotato di migliori capacità sonore grazie all'apposito circuito SID e inoltre consente di definire gli sprite per ottenere facilmente delle animazioni grafiche.

Il Plus 4, invece, ha una memoria RAM più estesa e 4 applicazioni (foglio elettronico, database, word processor e analisi grafica) integrate e sempre a disposizione. Inoltre offre alcune comode istruzioni BASIC relative allo sfruttamento delle possibilità grafiche e sonore.

Per questo motivo è impossibile utilizzare direttamente sul Plus 4 dei programmi concepiti per C 64, a meno di non effettuare una traduzione delle istruzioni non comuni ai due linguaggi.

Tenga però in conto che invece il Plus 4 è perfettamente compatibile con il C 16, per il quale ultimamente è stato reso disponibile parecchio software.

Anche **Personal Software** ha pubblicato (e continuerà a farlo) diversi programmi utilizzabili dai possessori di C 16 e Plus 4. Inoltre sono recentemente stati pubblicati dei libri dedicati espressamente a queste due macchine. Le ricordo, in particolare i volumi del Gruppo Editoriale Jackson e della *J.soft* che ha recentemente presentato un libro tutto di programmi per C 16 e Plus 4. ■

# Che cosa ha in

# Libreria Pr Personal Com

LITTLE TRAMP CHARACTER LICENSED BY BUBBLES, INC., S.A. GGK

# più il Personal Computer IBM?

ogrammi  
nputer IBM

## La Libreria Programmi Personal Computer IBM, per esempio.

Il tuo Concessionario IBM per il Personal Computer ha una novità per te: la Libreria Programmi Personal Computer IBM.

La Libreria Programmi Personal Computer IBM è uno strumento utilissimo, che ti permette di trovare nel modo più immediato i programmi firmati da IBM. Puoi guardarli e confrontarli per vedere subito, fra quelli di uno stesso settore applicativo, quale sia il programma che risponde meglio alle tue esigenze di lavoro.

Ma la Libreria Programmi Personal Computer IBM ti dà anche un aiuto in più: con ogni programma troverai la brochure che lo riguarda. Ce ne sono anche molte altre, che ti illustreranno tutto quello che il tuo Personal Computer IBM può fare e le sue applicazioni particolari, come il Videotel.

Puoi prenderle e portarle a casa, per leggerle in tutta tranquillità e pensare bene alla scelta che devi fare.

E non dimenticare che il Concessionario IBM (gli indirizzi sono sulle Pagine Gialle) è un vero esperto, che conosce perfettamente il Personal Computer IBM ed i problemi della tua attività. Potrà consigliarti nel modo migliore nella scelta dei programmi e ti illustrerà tutte le possibilità che il Personal Computer IBM ti offre, oltre alla grande versatilità delle sue prestazioni.

Per acquisto, consulenza e servizi, puoi anche rivolgerti al Negozio IBM Centromilano.

Allora, non ti pare che il Personal Computer IBM abbia veramente qualcosa in più?



Desidero ricevere:

- Informazioni sul Personal Computer IBM e i suoi programmi.
- Gli indirizzi dei Concessionari IBM Personal Computer della mia regione
- Una visita o dimostrazione pratica di un concessionario

Nome e Cognome.....

Azienda.....

La mia attività è.....

Indirizzo.....

Spedisci questo tagliando a: IBM Italia  
Direzione Entry Systems  
Casella Post. 137 - 20090 Segrate Milano

**IBM**

## Interfaccia programmabile per giocare con lo Spectrum

**U**no dei problemi di chi vuole utilizzare la macchina di Sir Clive per giocare, è la difficoltà di operare con i joystick a causa della mancanza di una porta dedicata a quest'aggeggio. Le soluzioni proposte, tra l'altro dalla stessa Sinclair con l'interfaccia ZX 2, non sono mancate, ma hanno generato una tale babele di "standard" contrastanti da rendere molto difficile la vita agli appassionati giocatori. La Frel Limited, società ovviamente inglese, ha risolto a suo modo il problema mettendo a punto ComCom, una piastra che consente di associare ai sei tasti scelti dal realizzatore del gioco per simulare l'effetto joystick (quattro direzionali e due per "sparare"), le effettive posizioni fisiche della magica manopola. È sufficiente, anche a gioco caricato, attuare una serie di immediate connessioni che rispecchiano le convenzioni del progettista sulla piastrina che comprende l'uscita per il joystick, ed il



gioco è fatto. Se il gioco prevede che si prema la lettera Q per ottenere lo spostamento laterale a sinistra, si connette il cavetto uscente dalla posizione segnata dalla freccia a sinistra con la posizione marcata Q sulla piastra. Questo hardware non disabilita la tastiera e rende quindi possibile agire anche con i giochi, come il Simulatore di volo, che richiedono la pressione contemporanea di più tasti. La scheda ComCom, di dimensioni (12 per 13,5 per 2 cm) e costo contenuti (meno di 20 sterline), ha un'uscita per i joystick e comprende un port d'espansione per l'eventuale connessione di sintetizzatori vocali e stampanti. La società distributrice possiede un ricco catalogo di accessori per macchine Sinclair e di altre marche, disponibile su richiesta scritta per i rivenditori interessati.

*Lion House (Retail) LTD  
International division  
227 Tottenham Court Road  
London W1P 0HX  
England  
Telex 28394 LION G*



## Confezione-raccoglitore per i floppy Memorex

**S**ensibile alle esigenze degli utenti di supporti magnetici, sempre alle prese con i problemi della conservazione dei dischetti flessibili, la Memorex ha messo a punto una soluzione relativamente originale ed economica affiancando alla classica scatola di cartoncino, una confezione da dieci floppy contenuti in una custodia rigida di materiale plastico. Questa custodia, che costa quanto un dischetto in più, permette di conservare i dischi in poco spazio e offre una discreta comodità per quanto riguarda l'uso dei supporti magnetici. Il coperchio viene parzialmente ribaltato per fungere da supporto alla scatola che può restare così aperta sul tavolo di lavoro, facilitando la visione dei dischi senza doverli estrarre. Vantaggi rispetto ad altre confezioni? Rigidità totale, costo limitato, praticità d'uso e ingombro ridotto. Il tutto per raggiungere l'obiettivo dell'alta affidabilità e sicu-



rezza nell'uso dei supporti, per contribuire all'integrità dei dati.

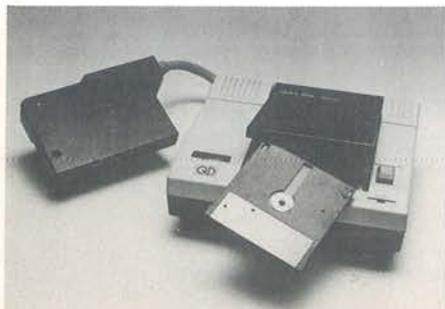
*Memorex Italia*  
Via C. Menotti, 14  
20129 Milano  
Tel. 02-718551

su ogni supporto. Il sistema operativo, ampliato con una serie di comandi che permettono l'accesso diretto a dati e programmi residenti su disco, non "brucia" la RAM riservata alle applicazioni. Alcuni di questi comandi aggiuntivi vengono associati ai tasti funzione per facilitare la vita all'utente: è in ogni caso possibile richiamarli esplicitamente mediante una serie di opportune Call. Con una velocità di rotazione di 423 giri al minuto e un tempo di trasferimento intorno ai 101 Kbit al secondo, le prestazioni dichiarate dalla casa per il Quick Disk Drive sembrano essere in grado di soddisfare le esigenze degli appassionati che desiderano passare dal registratore a cassette ad un supporto più comodo, senza però spendere "un occhio della testa". Le note informative parlano di un prezzo inferiore a quello di acquisto dell'home computer: è già qualcosa...

*Fowa S.p.A.*  
Via Tabacchi, 29  
10132 Torino  
Tel. 011-897373

## Quick Disk Drive per sistemi MSX

**L**a Yashica propone un nuovo accessorio per i computer che si rifanno allo standard MSX, già discretamente introdotto anche in Italia. È un lettore di dischi, il cui formato sconcerta parecchio, trattandosi dell'ennesima variazione sul tema intorno ai tre pollici. Non contenti delle ampie possibilità di scelta offerte dal mercato (3" - 3,25" - 3,5"), i progettisti della Yashica, già avvezzi a dilettersi di computer come nel caso dell'YC 64, hanno sfornato un lettore di dischi da 2,8", i più ridotti in questa fascia, capace di stipare 128 Kbyte



## Energia, un concetto in mostra

**È** partita in Febbraio da Bologna e ha raggiunto Roma in Maggio, per fermarsi fino a fine Luglio, una mostra che si propone di essere un tentativo per avvicinare il vasto pubblico a quel complesso di problemi che stanno intorno al concetto di energia. Realizzata dal comune di Bologna, Assessorato alla Cultura, in collaborazione con la Philips S.p.A., la mostra è divisa in tre sezioni: Energia e Materia (scienze fisiche), Energia e Vita (biologia) e Energia e Società (attività umane). A disposizione dei visitatori una videoteca con 60 film nazionali ed esteri che riprendono ed approfondiscono le tematiche presentate nell'esposizione. Il criterio partecipativo dell'esposizione viene esplicitato mediante esperimenti e dimostrazioni condotti da animatori e dalla possibilità di intervenire su dispositivi progettati ad hoc per la rappresentazione di alcuni fenomeni o princi-

pi. Il contributo della Philips, presente fin dalle fasi di impostazione del lavoro, è rappresentato anche da una serie di oggetti prodotti dalla casa di Eindhoven e progettati in Europa: elettrodomestici, sistemi di illuminazione, personale e home computer, applicazioni, dal laser alla musica (compact disc) e all'immagine (videodisco).

*Comune di Bologna*  
Assessorato alla Cultura  
Via Oberdan, 24  
40126 Bologna  
Tel. 051-336776



## Mettiamo un po' d'ordine!

**D**all'Inghilterra giunge un accessorio semplice, un uovo di Colombo, capace di sollevare da piccoli fastidi un esercito di amatori alle prese con ... le prese. È un



selettore a 3 ingressi e un'uscita che consente di collegare in permanenza ad un televisore fino a tre apparecchi ausiliari, scegliendo poi, per mezzo di un tasto, l'input desiderato in un certo istante. È possi-

bile così connettere stabilmente ad un solo apparecchio TV l'antenna per la ricezione dei programmi, un computer ed una consolle per videogiochi e scegliere poi comodamente quale collegamento utilizzare senza doversi districare in una marea di cavi. Garantito per 5 anni, MasterSwitch costa 4,86 sterline, VAT esclusa, è prodotto dalla Masterpiece e distribuito per l'estero da:

*ML Services*  
33 Shelgate Road  
London SW11 1BA  
England

### Interfaccia per radiocomunicazioni su C 64

**R**adiomatori udite, udite! La scheda Com-In 64 trasforma il Commodore 64 in un sistema terminale per radiocomunicazioni in Baudot, Morse, ASCII e SSTV. Oltre a ciò è possibile usare la scheda come modem telefonico, trasmettere e ricevere programmi BASIC e usare una sezione del programma come word processor. È sufficiente connettere l'interfaccia al computer, accenderlo e lavorare usando i 60 comandi previsti. È possibile usare anche il registratore a cassette o il disk drive per registrare dati e una stampante standard Commodore per averne una copia scritta. La flessibilità dell'hardware consente una precisa taratura dei filtri e di tutte le caratteristiche della rice-trasmissione utilizzando semplici istruzioni.

*Distribuzione:*  
*Bit Computers*  
Piazza S. Michele, 9  
17031 Albenga (SV)  
Tel. 0182-53512

*ATW Studio S.n.c.*  
Via Pestagalli, 7  
20138 Milano  
Tel. 02-502204

### Gestione archivio fotografico per PC

**L**o studio di Roberto Villa, per la distribuzione della New Reversal Service, ha realizzato un pacchetto per la gestione di archivi delle foto per amatori e, soprattutto, professionisti. È prevista una completa schedatura di ogni immagine e la possibilità di ricerca consente di impostare più condizioni contemporaneamente. Gli acquirenti del package, che può essere eseguito indifferentemente su macchine con floppy o hard disk, riceveranno in omaggio una serie di programmi autodidattici e di utilità.

*Audiovisual Communication*  
Via Altino, 3  
20144 Milano  
Tel. 02-496233

### Rotocalco radiofonico d'informatica

**V**a in onda su Radio Milano International, la gloriosa radio libera che ha recentemente festeggiato il suo ennesimo compleanno. "Computermania", una trasmissione dedicata al piccolo oggetto della rivoluzione elettronica. Presentata dal computer shop Logical Station 3001, la trasmissione è condotta da due giornalisti specializzati e tratta amichevolmente di tutto ciò che riguarda il mondo della piccola informatica, ogni lunedì dalle 14,40 alle 15 sui 98,100 e sui 101 MHz. Le rubriche sono dedicate alle novità hardware e software, ai linguaggi di programmazione, ai videogiochi ed ai programmi professionali e vengono intervallate da brani musicali suonati al calcolatore. Completano la trasmissione interviste a personaggi famosi e sconosciuti, una hit parade settimanale

del software più venduto e una serie di aneddoti e curiosità.

*Logical Station 3001*  
Piazza S. Maria Beltrade, 8  
20121 Milano  
Tel. 02-867935

### Programmi didattici per Spectrum

**U**n gruppo di docenti della Facoltà di Ingegneria dell'Università di Ancona ha prodotto una serie di programmi didattici per lo studio della chimica, che ora distribuisce per una valutazione (dietro rimborso delle spese postali e del supporto) a studenti ed insegnanti, ad esaurimento della scorta di 100 pezzi previsti. Realizzati per Spectrum 48 Kbyte RAM, i programmi sono rivolti a studenti delle scuole superiori e del primo anno di università. Il supporto di distribuzione è la cassetta che contiene una dozzina di "pezzi" che spaziano dall'insegnamento dei simboli e delle valenze alla titolazione. L'autore ringrazia quanti vorranno collaborare, anche con suggerimenti e critiche, e prega di allegare il francobollo per la risposta.

*Dott. Liberato Cardellini*  
Facoltà di Ingegneria  
Via della Montagnola, 30  
60128 Ancona  
Tel. 071-58931

# MISSIONE AFRICA

MTR

spazio gratuito offerto dal Gruppo Editoriale Jackson

**Una "Nave della Pace" in partenza dall'Italia porterà in Africa soccorsi immediati e aiuti per un domani migliore con il vostro contributo.**



La "Nave della Pace": uno strumento nuovo per intervenire immediatamente là dove è necessario un urgente soccorso a popolazioni gravemente colpite da calamità naturali. Non porta solo aiuti immediati ma anche strumenti di lavoro e mezzi ausiliari idonei a realizzare migliori condizioni di vita.



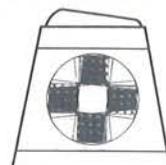
Affinché questo risultato sia raggiunto, gli aiuti siano adeguati ed efficaci e la "Nave della Pace" possa partire a pieno carico occorrono offerte e solidarietà da parte di tutti. Specialisti dei pro-

blemi dello sviluppo hanno offerto consigli per la migliore attuazione del progetto. L'iniziativa ha ottenuto l'Alto Patronato del Presidente della Repubblica Italiana, il

Patrocinio dell'ONU, della Lega di Società di Croce Rossa e di Mezza Luna Rossa e della Croce Rossa Italiana e il

contributo del Dipartimento per la Cooperazione allo Sviluppo del Ministero degli Affari Esteri. Il Comitato "Nave della Pace" chiede a quanti comprendono la necessità ed il valore dell'iniziativa, offerte di

mezzi e di beni indispensabili per i soccorsi di prima necessità e per gli aiuti di sviluppo destinati a 15 paesi dell'Africa.



**Nave della Pace**

COMITATO NAVE DELLA PACE - VIALE MAZZINI 41  
00195 ROMA - TEL. 06/317447-386163

Da compilare in stampatello ed inviare in busta chiusa a:  
Comitato Nave della Pace  
V.le Mazzini 41 - 00195 Roma.

**SI', ANCH'IO VOGLIO AIUTARE LA NAVE DELLA PACE A PARTIRE CON LE STIVE PIENE**

Cognome \_\_\_\_\_

Nome \_\_\_\_\_

Via \_\_\_\_\_ N. \_\_\_\_\_

CAP \_\_\_\_\_ Località \_\_\_\_\_

Per questo ho deciso di inviare il mio contributo di

- Lit. 10.000  Lit. 50.000  
 Lit. 25.000  Lit. 100.000 o più

tramite:

Assegno non trasferibile intestato: Comitato Nave della Pace

C/c postale n. 15285000

Bonifico bancario a credito del c/c n. 3100/51 c/o la Cassa di Risparmio di Roma sede centrale -

Via del Corso, 320 - 00186 Roma

Desidero una ricevuta del mio versamento.



**H**a inizio da questo numero una serie di articoli sulla simulazione dinamica realizzata mediante il computer. L'argomento è di grande interesse perché fornirà al lettore lo strumento per studiare la realtà che lo circonda, riproducendola "in vitro" nel proprio calcolatore; potranno in questa maniera essere studiati fenomeni fisici, sistemi biologici, reazioni chimiche, impianti dotati di regolazione automatica.

Non essendo vincolati dalla materia, potremo realizzare esperimenti senza dover disporre di un laboratorio, di tecnici e delle apparecchiature necessarie; per quel che riguarda, poi, gli impianti, si vedrà come è possibile collaudarne l'efficienza e la funzionalità prima di averli costruiti.

Per il tono divulgativo degli articoli, rivolti a tutti e non soltanto a degli specialisti, si dovranno necessariamente introdurre delle approssimazioni e delle imprecisioni. D'altronde l'obiettivo che si vuole conseguire è soltanto di trasmettere delle idee e dei concetti di base; si rimanda quindi ai testi specifici per una trattazione più rigorosa.

Sulla simulazione al calcolatore sono già stati pubblicati degli articoli su **Personal Software** (n. 10-11 e n. 22). Questa volta entreranno più in dettaglio descrivendo anche le tecniche da usare e ricorrendo a numerosi esempi.

La simulazione al calcolatore può essere statica o dinamica, per unità discrete o nel continuo, e infine deterministica o stocastica; senza farci prendere dal panico, esaminiamo cosa vuol dire tutto ciò.

La simulazione statica si riferisce a fe-

nomeni che non si evolvono nel tempo; la simulazione dinamica viceversa segue l'evolversi del fenomeno nel tempo, e quindi lo visualizza ad intervalli regolari per tutta la durata della simulazione. Si dice che la simulazione viene fatta per unità discrete quando l'argomento del fenomeno da studiare consiste in entità ben definite: persone in attesa ad uno

# La simulazione dinamica di fenomeni continui

## Un uso intelligente del calcolatore per lo studio della realtà

di Franco Sardo

sportello, auto in fila, pezzi in una catena di montaggio. Quando invece l'argomento è una grandezza fisica che varia nel tempo con continuità, come la velocità di un corpo che cade o la temperatura di un cappuccino che si raffredda, allora la simulazione è nel continuo.

La realtà si evolve nel tempo, quasi mai è statica. Un fenomeno o un esperimento deve essere simulato nel suo svolgimento temporale, e le grandezze in esame variano con continuità; noi quindi parleremo di simulazioni *dinamiche nel continuo*.

Nell'effettuare un esperimento, possono intervenire fattori esterni casuali, come le variazioni della temperatura atmosferica, o il guasto di una macchina. Se si introducono nella simulazione questi

zioni partendo dagli stessi valori iniziali e con le stesse leggi otterremo lo stesso risultato. In questo caso il modello si dice *deterministico*.

Vediamo adesso quali sono le applicazioni pratiche dei programmi di simulazione: abbiamo già parlato della possibilità di effettuare esperimenti fisici senza disporre delle attrezzature necessarie; diremo inoltre che è possibile, mediante la simulazione, effettuare esperimenti che non potrebbero essere realizzati altrimenti.

Ad esempio, quale sarebbe il moto della Terra se Giove avesse una massa doppia di quella che ha? La risposta si può avere analiticamente mediante le leggi di Keplero e di Newton, ma se si vuole vedere l'effetto sperimentale è ovvio che bisognerà ricorrere alla simulazione. Altrettanto dicasi per reazioni nucleari o

fattori casuali, mediante un opportuno utilizzo della funzione Rnd, in modo tale che ogni volta che si fa girare il programma vengano fuori risultati diversi, la simulazione si dice *stocastica*, perché stocastiche sono dette le variabili casuali.

Se viceversa non esiste nell'esperimento o nel fenomeno alcun fattore di casualità, ogni volta che ripeteremo la simula-

per il comportamento della materia nelle spaventose condizioni di gravità e densità che si riscontrano in un buco nero.

La simulazione è dunque un'alternativa al calcolo analitico, alternativa cui si ricorre quando il calcolo è troppo complesso, o quando si vuole seguire il feno-

## COMMODORE 64

meno nel suo evolversi e non in un momento particolare. Il calcolo analitico si può paragonare a una foto, la simulazione ad un filmato.

D'altronde si sta facendo strada presso la comunità scientifica la convinzione che esistono dei fenomeni che non possono essere descritti da leggi fisiche, ma solo dalla simulazione. La legge fisica è in questi casi sostituita da un algoritmo risolutivo.

Mediante la simulazione si possono inoltre verificare sistemi di regolazione automatica, naturali o costruiti dall'uomo; si controllerà, simulandoli, che essi riescano realmente a mantenere le condizioni richieste senza pendolazioni; si può arrivare, come si diceva prima, a verificare la funzionalità di un impianto, cioè a collaudarlo, prima di averlo costruito.

Tutto ciò presenta delle difficoltà tecniche che sono state in passato oggetto di studi approfonditi. In particolare, la simulazione dinamica di fenomeni continui cozza contro difficoltà matematiche considerevoli. Ciò perché la realtà che si evolve nel tempo è descritta da equazioni differenziali, cioè da equazioni che mostrano la variazione (in questo caso nel tempo, istante per istante) della grandezza in esame.

Esse fanno parte di una branca della matematica che si chiama calcolo infinitesimale, introdotto a suo tempo da Newton e Leibniz.

Alcuni di noi saranno probabilmente in grado di risolvere un'equazione differenziale, ma purtroppo ogni fenomeno fisico coinvolge di solito più variabili, ed è pertanto descritto da un sistema di più equazioni differenziali; si arriva così facilmente a livelli di difficoltà tali da met-

tere in crisi il più preparato dei matematici.

Ad esempio, un corpo che cade nell'aria varia continuamente la propria velocità, secondo la seguente legge:

$$dV/dT = g - KV^{12}$$

Questa equazione dà la variazione infinitesima di velocità  $dV$  nel tempo  $dT$ . La

espressione  $dV/dT$  è la derivata della velocità rispetto al tempo, e viene chiamata accelerazione.

Non possiamo scrivere:

$$(V1 - V0)/(T1 - T0) = g - KV^{12}$$

considerando una variazione finita di velocità in un intervallo di tempo finito; ciò sarebbe lecito solo se la velocità variasse linearmente entro l'intervallo di tempo considerato; cosa che non è, a meno di non considerare un intervallo di tempo infinitamente piccolo.

La risoluzione di un'equazione differenziale di questi tipo non presenta problemi, e noi possiamo sapere facilmente quale sarà la velocità di un corpo dopo tre secondi di caduta.

Ma, come si diceva, nei problemi che si presentano nella realtà le variabili sono numerose, e le equazioni complesse, tanto da rendere impossibile la soluzione analitica.

Occorre allora mettere a punto un sistema che permetta di studiare le variazioni delle grandezze, segnatamente nel tempo, con metodi numerici, magari approssimati, ma tuttavia accettabili.

Poiché quindi risolveremo i nostri problemi non teoricamente, ma sperimentalmente con metodi numerici, potremo a buon diritto parlare di matematica

sperimentale.

Nel nostro studio della simulazione dinamica cominciamo quindi a mettere a punto questo metodo: e lo faremo utilizzando l'esempio dell'oggetto che cade nell'aria.

Nell'equazione riportata si vede che l'aumento della velocità al passare del tempo è direttamente proporzionale a 9 (accelerazione di gravità, pari a 9,81 m/sec per secondo) e inversamente proporzionale al quadrato della velocità a causa dell'aumentare, con la velocità stessa, della resistenza dell'aria.

$K$  è una costante di proporzionalità che dipende da vari fattori, e che a titolo esemplificativo porremo pari a 0,006. Il termine  $KV^{12}$  ha il segno meno perché

provoca una diminuzione della velocità. Chiamiamo  $A$  questa variazione di velocità  $dV/dT$  (accelerazione); avremo:

$$A = g - KV^{12}$$

Questa viene chiamata equazione di Tasso, perché ci dà il tasso di variazione della grandezza  $V$ .

Se la velocità iniziale è  $V0$ , la velocità dopo un piccolissimo intervallo di tempo sarà:

$$V1 = V0 + AdT$$

cioè sarà aumentata del tasso di aumento istantaneo moltiplicato per l'intervallo di tempo considerato.

Poiché dopo questo intervallo la velocità si sarà stabilita ad un nuovo livello, come una bottiglia parzialmente piena in cui abbiamo aggiunto un po' d'acqua, questa viene chiamata equazione di Livello.

Se il  $dT$  fosse infinitamente piccolo, l'equazione sarebbe risolta in maniera esatta, ma ciò comporterebbe un numero infinito di calcoli; si considera allora, per una soluzione approssimata, un valore di  $dT$  piccolo ma finito; nel caso del corpo che cade, un valore adeguato potrebbe essere 0,01 secondi.

Se vogliamo allora sapere quale sarà la velocità dopo tre secondi di caduta, dovremo spezzettare questi tre secondi in 300 intervalli di 0,01 sec; in ogni intervallo calcolare il tasso di variazione della velocità  $A$ , e aumentare la velocità di un valore pari a questo tasso moltiplicato per l'intervallo considerato.

Nel calcolo dell'equazione di tasso useremo, come valore da sostituire in  $V^{12}$ , ogni volta il valore determinato dall'equazione di livello del ciclo precedente. Il livello di  $V$  continuerà quindi a crescere di valori piccolissimi.

Come si vede, è indispensabile conoscere le condizioni iniziali (cioè la veloci-

tà al tempo 0). Dovremo inoltre costruire un ciclo, (che si ripete N volte fino ad arrivare al tempo desiderato) in cui si alternano un'equazione di tasso ed una di livello.

Col metodo descritto si ottengono delle buone approssimazioni al valore esatto. Che l'approssimazione sia buona si verifica effettuando varie prove, ogni volta riducendo l'intervallo di tempo (detto intervallo di soluzione). Se continuando a ridurlo non si hanno variazioni apprezzabili nel valore della grandezza al tempo T (nel nostro caso nel valore della velocità dopo tre secondi), vuol dire che il valore ottenuto è abbastanza vicino a quello reale.

Risolvere un'equazione differenziale si dice integrarla; il procedimento descritto è un'integrazione alle differenze finite. Esistono altri sistemi di integrazione alle differenze finite più accurati, ma questo è senz'altro il più semplice.

Per chiarire quanto detto, realizziamo ora il programma per calcolare la velocità di caduta di un corpo nell'aria.

La struttura del programma sarà la seguente:

1 - introduzione dei valori delle costanti;  
2 - introduzione valori iniziali delle variabili;

3 - introduzione dell'intervallo di soluzione e della durata della simulazione;  
4 - ciclo con visualizzazione dei risultati intermedi;

5 - stampa dei risultati finali.

I punti 1, 2, 3 e 5 sono, secondo il linguaggio della programmazione strutturata, "action blocks", senza alcun problema.

Il punto 4 è un loop, che in BASIC va impostato così:

```
400 REM loop
410: T = T + dT
420: A = g - k * V2
430: V = V + A * dT
440 IF T > TMAX THEN 500:REM
      condizioni di uscita
450 GO TO 400
500 REM end loop
```

La linea 410 è il contatore del tempo, essenziale in ogni programma di simulazione dinamica.

La linea 420 è l'equazione di tasso, che calcola il tasso di variazione della velocità nell'intervallo di tempo considerato,

e la linea 430 è l'equazione di livello, che modifica il livello della grandezza in esame, nella fattispecie la velocità. L'intero programma sarà allora quello riportato nel listato 1.

Il programma visualizza l'andamento della velocità nel tempo, e stampa il suo valore alla fine del periodo considerato. Disponiamo adesso di uno strumento per studiare l'evolversi dei fenomeni nel tempo. Abbiamo realizzato una simulazione estremamente semplice, ma lo stesso metodo potrà essere applicato a simulazioni più complesse: sarà sufficiente conoscere le equazioni che descrivono il sistema da studiare, cioè il suo modello matematico.

Abbiamo visto come i fenomeni fisici e la realtà in generale non siano facili da simulare dinamicamente, perché il loro evolversi nel tempo è descritto da sistemi di equazioni differenziali.

Abbiamo allora risolto una equazione differenziale con un metodo alle differenze finite, mediante equazioni di tasso ed equazioni di livello, e con ciò abbiamo simulato la caduta di un corpo nell'aria.

#### TABELLA DI CONVERSIONE

```
{HOME}.....HOME
{CLR}.....PULIZIA SCHERMO
{CUR.SU}.....CURSORE IN ALTO
{CUR.GIU}.....CURSORE IN BASSO
{CUR.DES}.....CURSORE A DESTRA
{CUR.SIN}.....CURSORE A SINISTRA
{SPC}.....SPAZIO
{RVS ON}.....REVERSE ON
{RVS OFF}.....REVERSE OFF
{INST}.....INSERT
{F1}.....TASTO F1
{F2}.....TASTO F2
{F3}.....TASTO F3
{F4}.....TASTO F4
{F5}.....TASTO F5
{F6}.....TASTO F6
{F7}.....TASTO F7
{F8}.....TASTO F8
{BLACK}.....COL. NERO (CTRL+1)
{WHITE}.....COL. BIANCO (CTRL+2)
{RED}.....COL. ROSSO (CTRL+3)
```

```
{CYAN}.....COL. CIANO (CTRL+4)
{PURPLE}.....COL. PORPORA (CTRL+5)
{GREEN}.....COL. VERDE (CTRL+6)
{BLUE}.....COL. BLU (CTRL+7)
{YELLOW}.....COL. GIALLO (CTRL+8)
{ORANGE}.....COL. ARANCIO (CBM+1)
{BROWN}.....COL. MARRONE (CBM+2)
{LT.RED}.....COL. ROSSO CHIARO (CBM+3)
{GRAY1}.....COL. GRIGIO 1 (CBM+4)
{GRAY2}.....COL. GRIGIO 2 (CBM+5)
{LT.GREEN}.....COL. VERDE CHIARO (CBM+6)
{LT.BLUE}.....COL. BLU CHIARO (CBM+7)
{GRAY3}.....COL. GRIGIO 3 (CBM+8)
```

I CARATTERI GRAFICI, OTTENUTI CON LA PRESSIONE DEI TASTI 'SHIFT' E 'CBM', SONO CODIFICATI IN MODO DA INDICARE IL TASTO DA PREMERE ASSIEME A 'SHIFT' O 'CBM'. ES. IL CUORICINO E' CODIFICATO CON {SH S}.

UN NUMERO DENTRO LE PARENTESI INDICA LE VOLTE CHE IL TASTO VA PREMUTO.

## COMMODORE 64

Listato 1 - Il programma che visualizza l'andamento della velocità nel tempo.

```

100 REM *****
110 REM *   SOLUZIONE DI EQUAZIONI   *
120 REM *   DIFFERENZIALI           *
130 REM *****
140 :
150 :
160 REM COSTANTI
170 REM -----
180 G=9.81:K=0.006
190 :
200 REM DURATA
210 REM -----
220 PRINT"{CLR}":PRINT"DURATA DELLA SIMU
    LAZIONE:";
230 INPUT TMAX
240 :
250 REM INTERVALLO DI SOLUZIONE
260 REM -----
270 PRINT:PRINT"INTERVALLO DI SOLUZIONE:

```

```

";
280 INPUT DT
290 :
300 REM CONDIZIONI INIZIALI
310 REM -----
320 V=0:T=0
330 :
400 REM LOOP
410 REM ----
420 :T=T+DT :REM CONTATORE DEL TEMPO
430 :R=G-K*V*V:REM EQUAZIONE DI TASSO
440 :V=V+R*DT :REM EQUAZIONE DI LIVELLO
450 :IF T >= TMAX THEN 500
460 GOTO 400
470 :
500 REM STAMPE FINALI
510 REM -----
520 PRINT"{5 CUR.GIU}TEMPO,SEC:";INT(T*1
    00)/100
530 PRINT"{CUR.GIU}VELOCITA',M/SEC:";INT
    (V*100)/100
540 GET A$:IF A$="" THEN 540

```

Vediamo adesso di applicare questo metodo a simulazioni più complesse. Immaginiamo un recipiente in cui avviene una reazione chimica (reattore); immaginiamo che la reazione sia esotermica e che il recipiente sia mantenuto a temperatura costante mediante acqua che passa entro una camicia di raffreddamento; il sistema è descritto:

- 1) dall'equazione cinetica che dà la velocità di reazione;
  - 2) dalla equazione che dà l'aumento della velocità di reazione all'aumentare della temperatura;
  - 3) dall'equazione che descrive la trasmissione del calore attraverso le pareti del recipiente;
  - 4) dall'equazione che descrive l'aumento di temperatura del liquido refrigerante fra ingresso e uscita della camicia.
- Tutto ciò senza tener conto della agitazione, delle dispersioni di calore verso l'esterno (si considera il recipiente perfettamente coibentato) e di tanti altri fattori.

Malgrado le semplificazioni introdotte, abbiamo già quattro equazioni, che, comportando variazioni continue nel tempo, saranno equazioni differenziali.

Il sistema è dunque descritto da un modello matematico costituito da quattro equazioni. Se le conosciamo, e disponiamo di alcuni parametri che vi com-

paiono, quali il calore di reazione, i coefficienti di trasmissione del calore dei materiali impiegati, la superficie della camicia di raffreddamento, ecc., potremo simulare con sufficiente esattezza il sistema prima ancora di costruirlo; se dalla simulazione emerge che non si riesce a mantenere la temperatura entro limiti accettabili, dovremo ricorrere ad un altro sistema di raffreddamento, o ad un liquido refrigerante a temperatura

più bassa, o ad un'aggiunta graduale dei reagenti.

Così facendo, avremo sostituito la realtà col modello matematico della realtà, e tante più variabili avremo considerato, tanto più il modello sarà aderente alla realtà, e tanto più utile sarà la simulazione.

Una volta disponibile il modello, cioè note le equazioni che descrivono il sistema e i parametri lì presenti, il passaggio al programma di simulazione è quello già descritto: avremo una serie di equazioni di tasso e di equazioni di livello

alternate fra loro.

Nell'esempio in questione, avremo una temperatura iniziale e una concentrazione iniziale dei reagenti. Da questi livelli iniziali, mediante le equazioni di tasso, sapremo di quanto aumenta la temperatura nel minuscolo intervallo di tempo

considerato e di quanto diminuisce la concentrazione dei reagenti. La temperatura del reattore sarà dunque ad un nuovo livello; dalla temperatura del reattore e da quella del liquido di raffreddamento sapremo, mediante l'equazione di tasso che descrive la trasmissione del calore, la quantità di calore che passa al fluido di raffreddamento. La temperatura del fluido si stabilirà ad un nuovo livello, e cambierà di conseguenza la temperatura del reattore; a

questo punto si considera un nuovo intervallo di tempo e il ciclo ricomincia, con tutte le variabili al valore a cui erano rimaste alla fine del ciclo precedente.

La struttura generale del programma sarà quindi ancora una volta:

- 1) inserimento valori delle costanti;
- 2) inserimento valori iniziali delle variabili;
- 3) ciclo:
  - equazioni di tasso + equazioni di livello;
  - visualizzazione o stampa dei risultati

al tempo T;

- incremento del tempo.

Può succedere che il passaggio dalle equazioni di tasso alle equazioni di livello non sia immediato; cioè, ad esempio, dalla equazione di tasso avremo saputo che nell'intervallo di tempo considerato si sviluppano dalla reazione 12.500 Kcal. Il livello che deve essere incrementato è invece la temperatura del reattore; occorrerà allora un'equazione intermedia, che trasformi il calore espresso in Kcal nel corrispondente

**Listato 2 - Il programma per la simulazione del raffreddamento di un serbatoio.**

```

100 REM *****
105 REM * SIMULAZIONE RAFFREDDAMENTO *
115 REM * DI UN SERBATOIO *
120 REM *****
130 :
140 :
150 REM DATI
160 REM ****
170 TO =170: REM TEMP.INTERNA
180 TM =35 : REM TEMP.MEDIA FLUIDO RAF
FREDDAMENTO
190 KG = 35000:REM MASSA DA RAFFREDDARE
200 CS = 1 :REM CALORE SPECIFICO
210 S = 32 :REM SUPERFICIE DI SCAMBIO
215 PO =2200 :REM PORTATA A.DI TORRE
220 QT=0 :REM CALORE TOTALE SOTTRAT
TO AL SISTEMA
225 K=600 :REM COEFF.DI SCAMBIO TERM
ICO
227 MIN=0 :REM CONTATORE DEL TEMPO
228 :
229 :
230 REM INIZIALIZZAZ.STAMPA
231 REM *****
232 OPEN4,4
233 PRINT#4,"MIN{7 SPC}QORARIO{5 SPC}QMI
N{5 SPC}DT{7 SPC}TINT{6 SPC}T USCITA
{4 SPC}QTOT"
234 PRINT#4
235 :
240 :
250 REM EQUAZ. DI TASSO
260 REM *****
270 QH=K*S*(T0-T1)

```

```

280 :
290 :
300 REM EQUAZIONI INTERMEDIE
310 REM *****
320 QMIN=QH/60
330 DT= QMIN/(CS*KG)
340 TU=30+QMIN/PO
350 TM=(30+TU)/2
360 :
370 :
400 REM EQUAZIONI DI LIVELLO
410 REM *****
420 QT= QT+QMIN
430 TO=TO-DT
450 :
460 :
500 REM STAMPA
510 REM *****
520 PRINT#4,CHR$(16)CHR$(48)CHR$(0)MIN;
530 PRINT#4,CHR$(16)CHR$(49)CHR$(0)INT(Q
H);
540 PRINT#4,CHR$(16)CHR$(50)CHR$(0)INT(Q
MIN);
550 PRINT#4,CHR$(16)CHR$(51)CHR$(0)INT(D
T*100)/100;
560 PRINT#4,CHR$(16)CHR$(52)CHR$(0)INT(T
0);
570 PRINT#4,CHR$(16)CHR$(53)CHR$(0)INT(T
U);
580 PRINT#4,CHR$(16)CHR$(54)CHR$(0)INT(Q
T)
590 :
600 :
610 MIN=MIN+1
620 :
630 :
640 GOTO 250

```

aumento di temperatura. Basterà dividere il calore per la capacità termica del sistema; (sommatoria delle masse presenti ognuna moltiplicata per il suo calore specifico).

Ecco allora che nell'alternanza fra equazioni di tasso ed equazioni di livello possono inserirsi delle equazioni intermedie, che rendono i risultati in uscita dalle une accettabili come dati in entrata per le altre.

Inoltre, bisognerà essere cauti nella scelta dell'intervallo di soluzione, cioè del lasso di tempo simulato fra una serie di calcoli e la successiva.

Se il fenomeno è la caduta di un corpo nell'aria, un intervallo di soluzione di 0,1 secondi è troppo grande un intervallo di 0,01 sec è adeguato, perchè si tratta di un fenomeno che si evolve rapidamente.

Nel caso di una reazione chimica, un intervallo di un minuto può essere sufficiente, ma nel caso di reazioni biologiche molto lente (fermentazioni) possono essere sufficienti anche intervalli di un'ora.

Un intervallo troppo lungo porta a risultati grossolanamente approssimati, mentre un intervallo troppo corto porta a moli enormi di calcoli e quindi a tempi di elaborazione inaccettabili.

diametro delle tubazioni dell'acqua. Diciamo già che risolvere il problema analiticamente, senza ricorrere alla simulazione, non è alla portata di tutti, e inoltre richiederebbe una gran mole di calcoli. Secondo lo schema riportato prima, il programma ha inizio con l'inserimento delle costanti delle equazioni:

- massa da raffreddare, Kg: 35000 kg
- calore specifico, Cs: 1 cal/gr
- superficie scambiante, S: 32 mq

Da alcuni semplici calcoli, conoscendo i materiali impiegati, si determina il coefficiente di scambio termico, k:

$$k = 600 \text{ Kcal/h. mq. } ^\circ\text{C}$$

Si stabiliranno quindi i valori iniziali delle variabili:

- Temperatura interna, T0: 170°C
- Temperatura del fluido di raffreddamento, TM: 30°C

Per quel che riguarda la portata del fluido di raffreddamento, si parte da un'ipotesi, 2.200 lt/min. Se questa portata si rivelerà insufficiente nel corso della simulazione, potremo aumentarla in seguito.

Si pone quindi a zero il contatore del tempo, che chiameremo MIN (= minuti); useremo infatti un intervallo di soluzione di un minuto, cioè la situazione del sistema sarà aggiornata e visualizzata di minuto in minuto.

Porremo inoltre a zero un'altra variabile, Q<sub>tot</sub>, ove accumuleremo il calore disperso ad ogni ciclo.

Ciò ci servirà per verificare, al termine della simulazione, che il calore totale perso dal serbatoio per arrivare a 60°C corrisponda al valore teorico.

Finite le inizializzazioni, si introduce una linea per la stampa dei titoli in testa alla tabella che conterrà i risultati; questi verranno stampati ciclo per ciclo. Inizia quindi il ciclo.

Si parte con l'equazione di tasso: il calore disperso attraverso la camicia di raffreddamento sarà:

$$Q_h = KS(T_0 - T_1)$$

Poiché K ha dimensioni Kcal/h.mq.°C, il Q<sub>h</sub> sarà espresso in Kcal/h. L'equazione riportata è in forma finita, il che avrebbe senso solo se il calore disperso nell'unità di tempo fosse costante; nel nostro caso essa varia man mano che il reattore si raffredda, per cui dovremo usare intervalli di tempo molto piccoli; altrimenti avremmo dovuto scrivere, in forma differenziale:

$$dQ/dT = KS(T_0 - T_1)$$

Seguono le equazioni intermedie, come prima cosa bisogna conoscere il calore disperso in un minuto:

$$Q_{min} = Q_h/60$$

Poi bisogna sapere di quanto si abbassa la temperatura per effetto della dispersione di questo calore:

$$dT = Q_{min}/(Cs \star Kg)$$

come si diceva, Cs★Kg, Kg di massa che si raffredda moltiplicato per il suo calore specifico, è la capacità termica del sistema, e quindi dt è l'abbassamento di temperatura che si verifica in un minuto nel reattore.

Questo calore disperso di trasferisce al fluido della camicia, riscaldandolo:

T<sub>u</sub> (temperatura di uscita) = 30 (temperatura di ingresso) + Q<sub>min</sub>/p<sub>o</sub>

dove P<sub>o</sub> è la portata dell'acqua di raffreddamento, in litri/min.

Si badi bene che questa non è un'equazione di livello: il liquido, che entra sempre a 30°C, uscirà ad ogni ciclo ad una temperatura diversa, decrescente man mano che il reattore si raffredda. Non si ha quindi quel fenomeno di accumulo

tipico delle equazioni di livello, in cui ad ogni ciclo si aggiunge al valore precedente la variazione intercorsa durante il ciclo stesso.

La temperatura media del liquido all'interno della camicia sarà, in prima approssimazione:

$$TM = (30 + T_u)/2$$

Infine abbiamo le equazioni di livello; le grandezze di cui ci interessa studiare il

Bisogna partire da intervalli di soluzione piuttosto lunghi, e poi continuare ad accorciarli finché i risultati non presentano più variazioni significative.

Vedremo adesso, come esempio di costruzione di un modello matematico e del suo impiego per la simulazione, il raffreddamento da 170° a 60°C di un serbatoio pieno di un prodotto.

Il prodotto chimico deve essere raffreddato prima di mandarlo alle lavorazioni successive. A tale scopo si manda nella camicia del serbatoio acqua di raffreddamento a 30°C; determineremo quanto tempo e quanta acqua sono necessari, in modo da poter stabilire il ciclo di lavorazione e da poter determinare il

# COMMODORE C 64

livello sono due: la temperatura interna del reattore e il calore totale disperso. Scriveremo:

$$T_0 = T_0 - dT$$

$$Q_{tot} = Q_{tot} + Q_{min}$$

Ciò nell'intervallo di soluzione (un minuto) la temperatura del reattore sarà scesa di  $dT$  e il calore totale che il reattore ha disperso dall'inizio del raffreddamento sarà aumentato di  $Q_{min}$ .

Concluso il ciclo, si stampano i valori ottenuti:

● MIN: tempo trascorso dall'inizio della simulazione, in minuti;

- $Q_h$ : il calore che il reattore disperderebbe in un'ora, in Kcal/hr;
- $Q_{min}$ : il calore disperso in quel minuto, in Kcal/min;
- $dT$ : la variazione di temperatura avuta in quel minuto nel reattore;
- $T_0$ : la temperatura interna del reattore;
- $T_u$ : la temperatura di uscita dell'acqua di raffreddamento;
- $Q_{tot}$ : il calore totale disperso dal reattore dall'inizio del raffreddamento fino a quel momento.

Ciò fatto, si incrementa il contatore del tempo:

$$MIN = MIN + 1$$

e si ricomincia il ciclo, questa volta utilizzando come valori di entrata delle equazioni di tasso i valori di livello del ciclo precedente.

Si sono introdotte, con questa simulazione, alcune approssimazioni, particolarmente due:

● la prima, che tutto il calore disperso dalla massa nel reattore vada nell'acqua di raffreddamento; ciò è legittimo, perché si considera il reattore coibentato e quindi trascurabili le dispersioni nell'ambiente;

● la seconda, che il liquido di raffreddamento abbia un gradiente di temperatura uniforme fra ingresso e uscita. Come sua temperatura abbiamo infatti assunto la media aritmetica fra  $T$  di ingresso e  $T$  di uscita. Cosa inesatta, ma poiché questa differenza non è alta (al massimo 20°C) l'approssimazione introdotta non ha effetti rilevanti.

Dalla simulazione otterremo due informazioni essenziali per la progettazione

Figura 1 - Un esempio di simulazione più complessa.

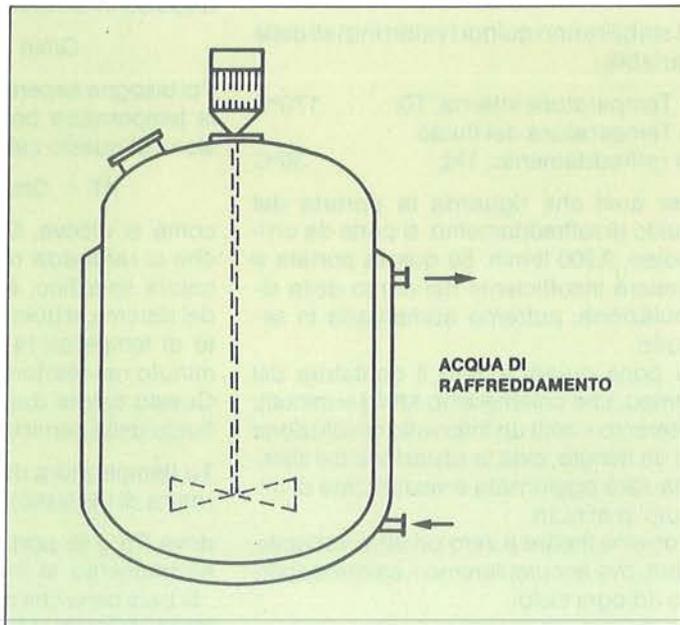
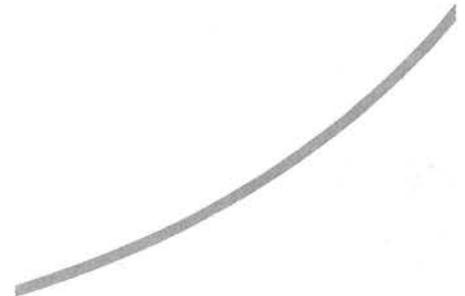
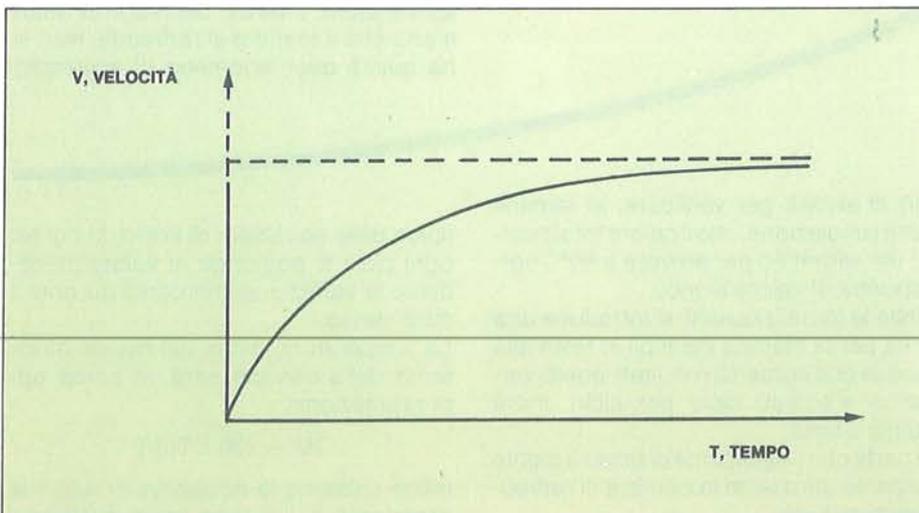


Figura 2 - Un grafico della velocità in funzione del tempo.

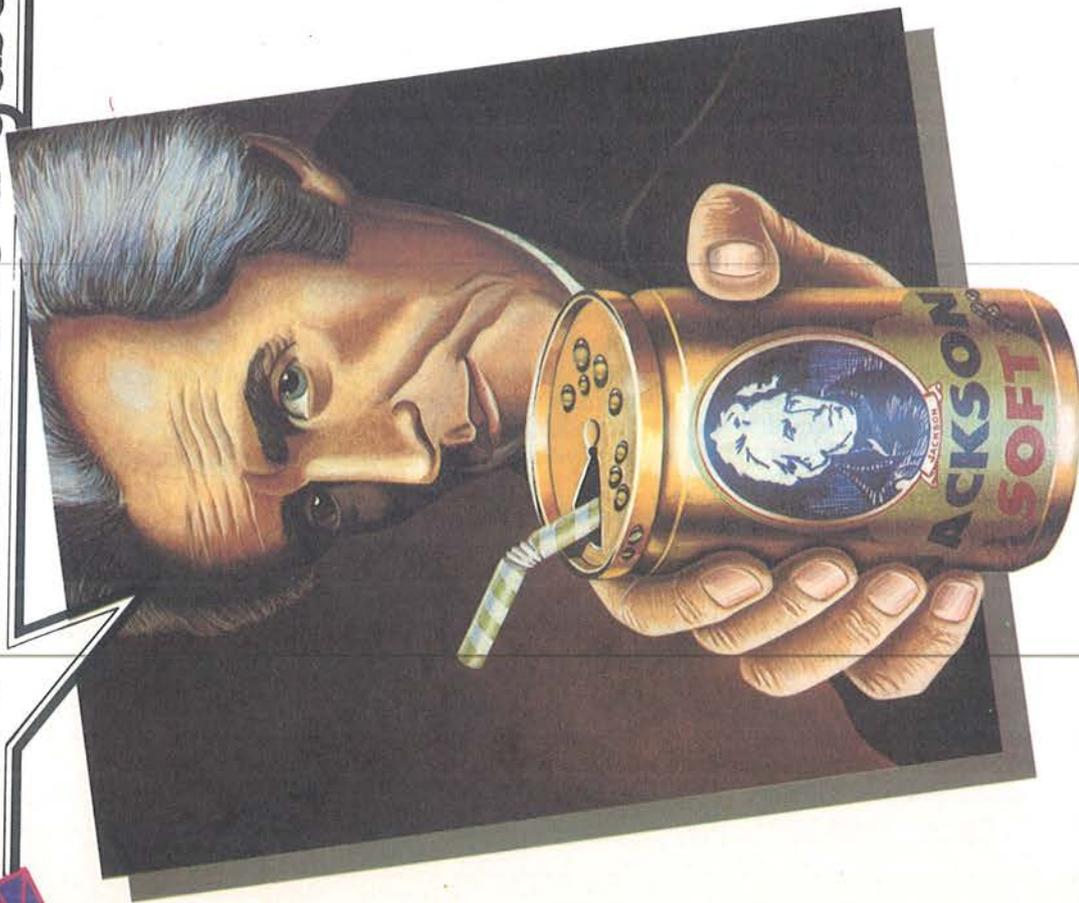


del sistema: in quanto tempo il reattore scende a 60°C (che è la temperatura che vogliamo) e a che temperatura esce l'acqua di raffreddamento in ogni minuto. Ciò è importante perché l'acqua di raffreddamento di solito proviene dalle torri evaporative, arriva sui 30°C e non la si può rimandare alle torri a temperatura superiore ai 40/45 °C.

Se quindi si verificasse tramite la simulazione che l'acqua esce dalla camicia a temperatura maggiore, occorrerà inevitabilmente aumentarne la portata. Infatti, la simulazione in questione portò alla decisione di usare 2.500 lt/min di acqua di raffreddamento invece dei 2.200 lt/min inizialmente previsti. ■

**FINALMENTE!**

**La Softrivista che ti gasa!**



**QUALCOSA DI SUPER, DI INEDITO,  
DI IRRESISTIBILE**

**IL VERO GIOCO  
COMINCIA ADESSO**

**IN EDICOLA  
JACKSON SOFT  
SERIE ORO**

I giochi esclusivi per Commodore 64 e Spectrum 48 K importati dall'Inghilterra, mai presentati in Italia. Una sfida Jackson al già visto, al già fatto, al... già registrato.



La prima puntata del fantastico, inedito **PYJAMARAMA**

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



GRUPPO EDITORIALE JACKSON

**I**l programma è previsto per computer PB-100, PB-200, e PB-300 e con qualche accorgimento su FX-702.

Normalmente un programma per gestione di dati, come quello di un'agenda telefonica, richiede una forte occupazione di memoria che pocket-computer come questi non sono in grado di gestire. Si pensi infatti che, nella biblioteca di programmi fornita all'acquisto del PB-100, viene fornito il listato di un programma che consente di registrare il numero di telefono e il nome di sole 14 persone. Aggiungendo la memoria di RAM (opzionale), è possibile registrare informazioni per 50/55 persone.

Il programma proposto vuole supplire a questa lacuna colmabile solo con lo sfruttamento di una memoria di massa esterna: tipicamente, un registratore a cassette.

Gli elementi hardware in questione sono: il computer PB-100 (espanso) o PB-200. L'interfaccia FA-3 per registrare, il registratore a nastro ed eventualmente la stampante FB-12.

Il programma si divide in 2 parti. La prima parte di registrazione fa memorizzare al nastro i dati interessanti secondo un certo formato. La seconda parte di "caricamento" si occupa della gestione degli stessi.

Riceve le informazioni necessarie, le cerca sul nastro, procede ai dovuti controlli e, infine si occupa di fornirli all'utente sull'unità periferica scelta: visore o stampante.

Il tempo di lavoro del sistema nella ricerca dei dati su cassetta è abbastanza alto. Ciò è dovuto al lento movimento del registratore e alla ricerca sequenziale dell'informazione. D'altronde questi pocket-computer non dispongono di drive o micro drive e per avvalerci di memorie esterne di massa non possiamo che usare esclusivamente dei registratori a cassetta.

Un secondo svantaggio è dovuto alle limitate capacità delle variabili di carattere, che possono immagazzinare solo 7 caratteri, fatta eccezione per la variabile di carattere esclusiva \$, che ne può contenere fino a 30.

Delegando proprio quest'ultima registrazione dei dati, dovremo dare a questa variabile un formato per consentire una facilità di gestione e un numero sufficiente di informazioni, entro natu-

# Agenda telefonica

## Un archivio con un pocket computer

di Fabio Fanecco

ralmente i 30 caratteri.

Come vantaggi si hanno:

- 1) la possibilità di aumentare notevolmente la capacità di gestione dei dati del pocket;
- 2) immagazzinare gli stessi in una semplice cassetta di pochissimo ingombro, non più a portata di mano di estranei, ma di computer;
- 3) una parte di programma che agevola la ricerca dei dati, lavorando autonomamente, evitando di farci sfogliare pagine di guide, agende, taccuini, con il rischio di farci confondere.

Esaminati i principali pro e contro, passiamo al funzionamento del programma.

Accesa l'unità con relativa interfaccia, il registratore e l'eventuale stampante, abbiamo subito la possibilità di scegliere tra le 2 modalità di lavoro:

- *Reg* che sta per registrazione dei dati;
- *Car* caricamento del o dei dati.

Digitando la lettera R entriamo nella fase di registrazione.

La prima domanda che ci viene posta riguarda il numero di input, cioè di dati, che vogliamo registrare. Impostato il numero voluto, ecco comparire l'ordine nel quale bisogna immettere i dati in una registrazione: Cognome, Nome N° telefonico, Città con la sigla, costituente la stringa di caratteri da registrare. Poiché

per ciascuna stringa si hanno a disposizione solo 30 caratteri, si dovrà scegliere e unificare un formato per permettere di immagazzinare il massimo numero di informazioni sufficienti e distinguibili, per poi estrapolarle al momento della richiesta.

A tal fine la stringa di caratteri viene divisa in 3 campi. I primi 19 caratteri contengono cognome e nome dell'interessato; i seguenti 8 caratteri sono riservati al numero di telefono. I restanti 3 invece, sono il campo destinato a contenere la sigla della città o, se preceduto dal simbolo ★, della provincia come riferimento all'eventuale prefisso telefonico (figura 1).

Il primo campo presenta un'ulteriore divisione. I primi 11 caratteri, infatti, sono destinati alla registrazione del cognome o di una sua abbreviazione, gli ulteriori 8 caratteri al nome. Il numero di telefono può spaziare tra i caratteri compresi tra il 19° e il 27°.

È importante registrare la sigla della città tra il 28° e il 30° carattere, sostituendo uno spazio al simbolo ★ se non occorresse. Una volta impostato e controllato il giusto formato, si posiziona il registratore nella modalità di registrazione; dopo aver digitato i dati sul computer, penserà il programma a trasferirli nel nastro, e a richiederci un nuovo dato.

Figura 1 - Suddivisione della stringa in 3 campi.

Cognome	Nome	6086481	★ MI
1	11	19	27
1° Campo Cognome e Nome		2° Campo N° telefono	3° Campo Sigla Città

# CASIO PB-100-200-300

Figura 2 - Alcuni esempi, così come vengono stampati.



È da sottolineare in questo caso l'utilità di poter disporre di un registratore munito di telecomando, che facilita notevolmente le operazioni di ingresso e uscita. La quantità di informazioni che si possono accumulare sono direttamente proporzionali alla durata della cassetta. Su una facciata di una cassetta C 60 siamo nell'ordine delle centinaia.

Terminata la parte di registrazione, passiamo alla parte di caricamento; se è stato rispettato scrupolosamente il formato dei dati, il computer sarà servizievole e preciso ad ogni nostra richiesta, anche se, come già accennato, impiegherà un discreto tempo nella ricerca dei dati.

Poniamoci quindi nella modalità "caricamento dei dati".

Il computer comincerà a porci una serie di domande. La prima riguarderà l'aiuto (Help), che per il momento tralascieremo rispondendo con un secco "No". La seconda riguarda l'emissione dei dati su video o, se vogliamo e ne siamo provvisti, sulla sua stampante.

Esplicate queste formalità, ci vengono richiesti i dati.

Prima ci viene chiesto di specificare il cognome dell'interessato. Anche per questa operazione bisognerà rispettare un formato, non superando gli 11 caratteri.

Passeremo poi a specificare il nome, senza superare gli 8 caratteri. Infine registreremo la città con la sigla preceduta da uno spazio, se l'interessato abita in quella città, o da un ★ se occorre ricercare un prefisso per una determinata provincia. Per esempio, se l'interessato abitasse a Saronno, occorrerà digitare ★ VA, poiché, Saronno è in provincia di Varese, e ricercare il relativo prefisso.

Posizionato il nastro e digitate le informazioni dovute, il computer si metterà all'opera cercando le informazioni richieste; i nominativi così trovati ci verranno mostrati sull'unità prescelta (visore o stampante).

Può darsi che nel fornire i dati per la ricerca, si sia incerti sulla città, sul nome, sul cognome, o che si vogliano più informazioni.

È a questo punto che interviene il programma di aiuto. Accettando l'aiuto, il computer ci mostrerà sul visore tutti i tipi di aiuto che può fornirci e i relativi codici di richiesta.

Il codice 1 è chiamato List All. Ciò ci consentirà di avere un aggiornamento di tutti i dati immagazzinati nella memoria, all'inizio della cassetta o a partire da un suo qualsiasi punto;

Il codice 2 Help: Cognome, fornisce all'utente il listato con le informazioni di tutti coloro che posseggono lo stesso cognome.

Se invece ricordiamo o conosciamo solo il nome, usufruiremo del codice 3 Help: Nome.

Il codice 4 Help: città, agglomera tutti coloro che abitano nella stessa città.

Per casi disperati infine si può ricorrere al codice 5, che identifica tutti coloro che posseggono la medesima lettera iniziale del cognome. Nulla quindi è mai perso!!!

Passando all'esame del listato si noterà che la sua struttura è assai versatile, prestandosi ad essere utilizzato per altri usi: catalogazione, gestione piccoli magazzini, prezzi, articoli di vendita ecc.

## REMARKS

La semplicità del programma è estrema.

Il programma P0 prevede lo smistamento ai vari sottoprogrammi a seconda del tipo di operazione che si voglia svolgere. L'input effettuato al passo 5 prevede un preciso valore alla variabile A\$, che viene esaminato al passo successivo. Infatti al passo 10 abbiamo la scelta If-Then che prevede o un passaggio al sottoprogramma 2 (caricamento dati), o al sottoprogramma 1 (registrazione dati).

Il programma P1 ha inizio con la registrazione al passo 1 del numero degli input che forniamo nel corso del programma che comanda al passo 10 un ciclo For-Next.

Ad ogni passo del ciclo la variabile esclusiva \$, su cui vengono registrati i dati Nome, Cognome Tel. Città, viene registrata sul nastro tramite l'istruzione Put \$ posizionata all'interno del ciclo stesso, al passo 20. Se vogliamo ricercare il dato, passiamo al programma P2. Dal passo 15 al passo 20 ha inizio la specifica dei dati richiesti: Cognome, Nome e Città.

Questi vengono rispettivamente immessi in 3 variabili di carattere: A\$, B\$, C\$.

Al passo 55 si procede ad estrarre una

# CASIO PB-100-200-300

qualsiasi variabile di carattere esclusiva \$, dalla quale al passo successivo verranno estratte con le stesse dimensioni fatte nell'operazione di input, 3 variabili: D\$,E\$,J\$.

Queste 3 variabili hanno gli stessi formati e hanno subito le stesse corrispondenze delle gemelle A\$,B\$,C\$.

Per sapere se la variabile \$ estratta contiene le informazioni ricercate, le sue sottostringhe DS,ES,JS, sono confrontate dal passo 65 al passo 75 con le corrispondenti A\$,B\$,C\$. Se i Nomi, Cognomi e le Città in esse contenute sono identiche, la variabile \$ viene ritenuta buona, e quindi accettata perché contenente ciò che a noi interessa. Al passo 80 l'istruzione Print \$ ce la mostrerà. In caso la variabile \$ non fosse quella, giusta, si ritorna al passo 55, in cui l'istruzione Get \$ preleva dal nastro un'altra variabile \$.

Il programma 4 è il programma di aiuto. Strutturalmente non differisce dal programma P2 per quanto riguarda la ricerca dei dati; la differenza sta nelle istruzioni di controllo. Al passo 10 infatti l'input del codice Help, di cui si è parlato precedentemente, definisce il tipo di estrazione di variabile.

Se S è uguale a 1 l'istruzione If-Then posta al passo 15 ci manda all'istruzione 43 che preleva incondizionatamente variabili \$ dal nastro, mostrandole tramite l'istruzione 50.

I successivi codici spostano l'esecuzione del programma in aree in cui i controlli delle variabili S sono effettuati in base al tipo di aiuto scelto.

In ogni caso vengono estratte delle sottostringhe dalla variabile esclusiva e confrontate con il dato originale.

Ad esempio l'istruzione 310 preleva dal Cognome scelto una stringa lunga 7

caratteri e posta nella variabile A\$.

Con l'istruzione Get \$ caricheremo dal nastro la variabile \$ letta e da questa estrarremo una stringa con lo stesso formato di quella A\$ che verrà usata per confronto, come si può notare dal passo 360.

Se il confronto risulta positivo, si dà via libera alla variabile S stampandone il contenuto e passando successivamente a caricarne un'altra con l'istruzione Goto 320 che il passo 350 impone.

Per concludere: i caratteri grafici, così come la scritta "List", sono inclusi in 2 sottoprogrammi ai passi 900 e 910 richiamabili all'occorrenza.

Listato 1 - Il programma Agenda telefonica.

```

P0          15 PRINT " FORNIRE          1 INPUT " VIDEO 0          60          605 GOSUB 900
1 PRINT "AGETEL"          I DATI: " ;          STAMPA (V/S)",          350 GOTO 320          610 GET $
2 VAC          20 INPUT " Cognome          Y$          360 PRINT $:GOSUB 9          620 A$=MID(1,1)
5 INPUT " REG.-CA          ",$:F=LEN($):Q          5 PRINT " 1 LIST          10          630 IF A$=B$ THEN 6
R.(R/C)",A$          =?:IF F<7:Q=F          ALL-2 HELP COGN          365 GOTO 320          50
10 IF A$="C" THEN          25 A$=MID(1,Q)          ONE";"-3 HELP N          400 INPUT " NOME ",          640 GOTO 610
#2          30 INPUT " Nome ",          ONE ";          $:F=LEN($):Q=7:          650 PRINT $:GOSUB 9
15 GOSUB #1          $:Z=7:F=LEN($):          IF F<7:Q=F          10          670 GOTO 610
16 END          IF F<7:Z=F          405 GOSUB 900          900 IF Y$="S":MODE
P1          35 B$=MID(1,2)          ER HELP ";          7:PRINT " ****
1 INPUT " N°INPUT          40 INPUT " Città'          10 INPUT " CODICE          420 GET $          * LIST ****
=" ,P          ",C$          HELP":,S          430 B$=MID(12,Q)          *:RETURN
5 PRINT "Cognome,          45 IF H$="S":MODE          15 IF S=1 THEN 43          440 IF A$=B$ THEN 4          910 PRINT "-----
Nome,Tel. ":"Ci          7          20 IF S=2 THEN 300          60          -----":R
tta'(SIGLA) ";          50 PRINT " **** AT          30 IF S=3 THEN 400          450 GOTO 420          ETURN
10 FOR I=1 TO P          TENDERE **** ";          40 IF S=4 THEN 500          460 PRINT $:GOSUB 9          10
15 PRINT I:"":IN          55 GET $          41 IF S=5 THEN 600          470 GOTO 420
PUT " DATI ",$          60 D$=MID(1,Q):E$=          43 GOSUB 900          500 INPUT " CITTA'"
20 PUT $          MID(12,Z):J$=MI          45 GET $          ,A$:GOSUB 900
25 NEXT I:RETURN          D(28,3)          50 PRINT $:GOSUB 9          510 GET $
P2          65 IF A$*D$ THEN 5          10          520 B$=MID(28,3)
1 INPUT " HELP (S          70 IF B$*E$ THEN 5          5          530 IF A$=B$ THEN 5
/N)",Y$          5          75 IF C$*J$ THEN 5          5          540 GOTO 510
5 IF Y$="S" THEN          80 PRINT $:MODE 8          90 END          550 PRINT $:GOSUB 9
#4          90 END          P4          10          560 GOTO 510
10 INPUT "VIDEO 0          305 GOSUB 900          310 A$=MID(1,Q)          600 INPUT " LETTERA
STAMPA (V/S)",H          =?:IF F<7:Q=F          320 GET $          INIZ.",B$
$

```

**I**mmagazzinare dati per poi operare sugli stessi nei più svariati modi è sempre stata l'applicazione regina nel settore informatico e anche lo sviluppo degli attuali personal e home computer non ha scalfito questa predisposizione degli elaboratori elettronici, non solo negli uffici, ma anche nell'ambiente "casalingo": chi non ha mai acquistato o si è copiato da una rivista o ha tentato di autocostruirsi un programma per la gestione di un certo quantitativo di dati?

Dopo questa premessa può sorgere spontanea la domanda del perché proporre qualcosa che tutti in qualche modo già possiedono. Il dubbio è più che plausibile, ma trova una risposta, spero valida, a due livelli: il computer sul quale è stato implementato il programma e lo scopo dello stesso. Il C 16, sia sul piano tecnico che economico, è indirizzato a un particolare settore di utenza, la cui principale caratteristica è di essere al primo approccio col mondo informatico e il cui primario interesse è di incominciare a muovere i primi passi. È in questo senso che si propone il presente programma: niente di impossibile o "professionale", ma un lavoro in grado di risolvere i dubbi e di presentare delle valide proposte didattiche a chi sta iniziando a percorrere la strada di una ricerca personale nell'ambito della programmazione di un home computer.

### Che cos'è un data base

Il primo dovere di un programmatore, anche se alle prime armi, è di aver ben chiaro il problema che vuole risolvere con la propria macchina, il che significa identificare le varie operazioni che dovranno essere eseguite e le connessioni logiche esistenti fra le stesse.

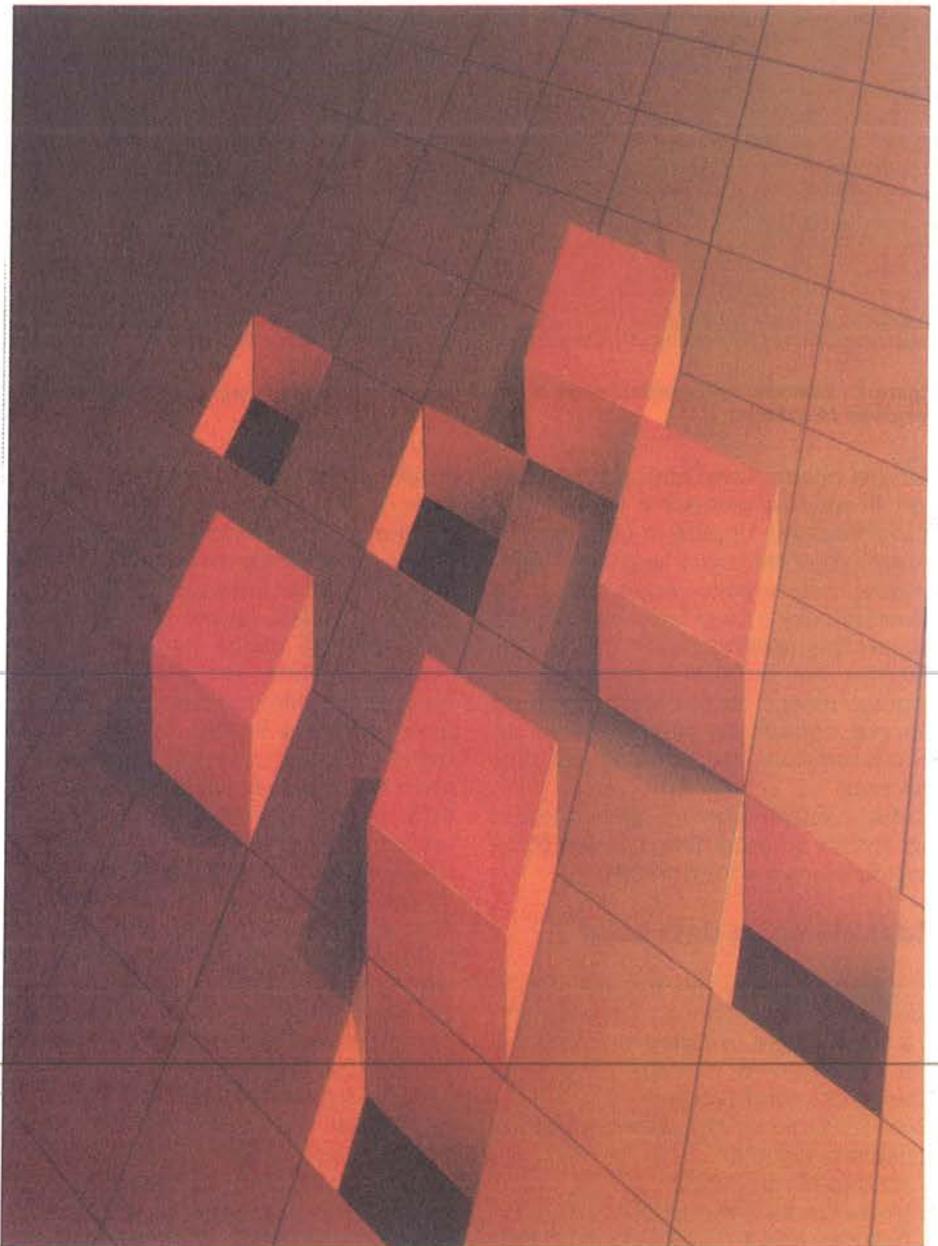
Lo sviluppo di un programma per la memorizzazione di dati può sembrare, a prima vista, una operazione abbastanza semplice in quanto richiede il ricorso ad un blocco di istruzioni quantitativamente minore rispetto, per esempio, anche ad un videogioco elementare. In effetti si tratta di utilizzare, prevalentemente, le istruzioni di assegnazione e di stampa, mentre non è in genere necessario ricorrere a routine grafiche, sonore o di movimento.

Ogni neoprogrammatore, superata l'impasse iniziale, è in grado, con poche

# Data base

Come archiviare ogni genere di dati in schedari su misura

di Claudio Poma



Input e Print e una variabile stringa con indice, di costruirsi un piccolo data base, ma è proprio a questo punto che dall'euforia si passa alla sconsolante presa di coscienza della stupidità della propria macchina e della difficoltà di metterla in grado di superare questo suo stato. L'esempio più lampante in questo senso è connesso all'incapacità del computer di operare per similitudini oltre che per identità. Nella pratica, si concretizza nel mancato ritrovamento dell'indirizzo di BIANCHI LUIGI perché i

- cambiare dei dati;
- cancellare dei dati;
- registrare i nuovi dati su un supporto esterno di memoria.

Oltre a questa suddivisione si pone ora il problema del contenuto dei dati che dovranno essere gestiti dal programma (indirizzi, libri, dischi, articoli di riviste, ricette, vini, collezioni di ogni tipo) in quanto ognuno di questi settori richiederà una diversa suddivisione dei campi del file.

Esistono due soluzioni al problema:

scheda, è cioè l'insieme di dati propri di un particolare soggetto o oggetto; mentre i campi sono i diversi argomenti trattati in ogni singola scheda. La figura 1, oltre ad esemplificare questa descrizione, permette di osservare la predisposizione del file a una gestione per matrice che il BASIC ci permette di organizzare con il semplice ricorso ad una variabile con indice.

## In teoria

### Introduzione di nuovi dati

L'operazione non presenta particolari problemi, ma alcune accortezze possono permettere un miglior sviluppo del listato e un risparmio pratico in memoria. Questo è ottenibile utilizzando una variabile con indice per i vari campi impiegati, il che permette di sintetizzare la routine di immissione con un'unico Input posto in un ciclo For-Next al posto di una lista di Input lunga quanti sono i campi trattati. Un tale procedimento potrà essere utilizzato vantaggiosamente anche nelle fasi di stampa dei dati dove un Print e un ciclo iterativo sostituiscono una lunga serie di Print.

Anche un controllo finale dei dati immessi, prima di una loro accettazione definitiva è importante, in quanto con tale opzione è possibile porre riparo immediatamente ad eventuali errori di distrazione commessi nella battitura dei testi. Non guasta anche un controllo della lunghezza dei dati dei diversi campi, soprattutto per problemi inerenti ad una successiva visualizzazione. Questa operazione sarà obbligatoria nel caso si sia deciso di impostare un programma che, per motivi di memoria RAM o memorizzazione su dischetti in file ad accesso casuale, richieda una limitazione particolare e predefinita.

Un altro problema che può insorgere in questa fase è il ricorso all'istruzione Input, molto sbrigativa, ma dagli effetti collaterali che a volte possono creare una situazione d'impasse: questo può succedere, per esempio, con la pressione, accidentale o meno, di tasti non richiesti (Run/Stop, frecce, ecc.). Per ovviare a questo inconveniente è senz'altro preferibile la simulazione di tale istruzione con il ricorso a Get, più macchinosa, ma dai risultati apprezzabili. Una tale impostazione risulterebbe ol-

	Campo n.1 NOME	Campo n.2 VIA	Campo n.3 CITTÀ	Campo n.4 TELEFONO
Record 1	Luigi	Via Verdi 4	Milano	67564534
Record 2	Sandro	Via Rossi 7	Milano	23435456
Record 3	Mario	Via Bianchi 3	Bergamo	5434234
Record 4	Paolo	Via Neri 45	Milano	88765601
Record 5	Marco	Via Gialli 9	Brescia	6473828

Figura 1 - Esempio dell'organizzazione di un file denominato Indirizzi organizzato in 4 campi e composto da 5 record.

dati del nostro, sono stati memorizzati con il nome a precedere il cognome (Luigi Bianchi), oppure in caratteri minuscoli (bianchi luigi o luigi bianchi) o, ancora, con le sole iniziali maiuscole (Bianchi Luigi o Luigi Bianchi), per non parlare di altre difficoltà (più di uno spazio tra i dati, errori di ortografia) che, seppur meno riscontrabili, possono sempre, comunque, contribuire ad un funzionamento non corretto del programma.

Vediamo quindi come si deve operare per fornire un programma di una struttura funzionale e funzionante.

## Costruiamo un data base

La prima cosa da fare è la definizione delle operazioni che il nostro programma deve essere in grado di compiere, per poi decidere il modo in cui ogni operazione deve svolgersi.

Dato l'argomento che stiamo trattando avremo bisogno di:

- introdurre nuovi dati;
- visualizzare i dati memorizzati;
- ricercare un particolare blocco di dati;

predisporre un programma specializzato su un particolare contenuto, o organizzare le cose in modo da poter aprire con lo stesso programma più schedari per gli usi più disparati. La soluzione scelta in questo lavoro è la seconda, senz'altro più valida per le maggiori possibilità offerte, anche se presenta il non trascurabile inconveniente di occupare molto più spazio nei chip silicei rispetto al primo caso; problema non da poco per una macchina come il C 16 dalle possibilità non certo illimitate in fatto di memoria.

Prima di passare alla definizione particolareggiata dei problemi connessi alle diverse operazioni necessarie a un data base che si rispetti, una breve digressione per i neofiti sul significato di tre vocaboli essenziali ad una corretta comprensione di quanto stiamo dicendo: file, record e campo.

Per file si intende l'insieme di tutti i dati riguardanti un particolare argomento (indirizzi, ricette, ecc.), è in pratica lo schedario che tanto si usa ancora negli uffici; il record, per rimanere nella simbologia utilizzata, corrisponde alla

## COMMODORE 16

tremodo vantaggiosa nella fase di correzione degli errori, come vedremo durante la descrizione di tale opzione.

L'unico punto che non può essere invece omesso durante la fase di introduzione dei nuovi dati è il controllo della memoria rimasta disponibile, al fine di non bloccare il programma per il superamento del dimensionamento scelto inizialmente o per l'occupazione di un numero di byte superiore a quelli disponibili.

#### Visualizzazione

È utilizzata soprattutto come subroutine dalle altre operazioni. Una particolare autonomia di questa funzione può avere dei vantaggi in due occasioni: la prima si ha nel caso che il numero di dati gestiti non sia molto elevato, in quanto, in tale situazione è possibile avere un certo controllo sul blocco dei dati nella sua globalità senza dover ricorrere alle altre operazioni. La seconda è data dall'eventuale bisogno di avere stampati su carta tutti i dati a disposizione.

Le modalità di intervento, sulla strutturazione del listato, varieranno nei due casi in relazione alle diverse istruzioni da utilizzare per il ricorso a video o a stampante, ma in generale l'espletamento di tale routine non presenta problemi di programmazione: un ciclo con contatore (due nel caso di una variabile con indice a due dimensioni) e l'impiego di un semplice Print.

#### Ricerca

Stiamo per entrare nel cuore del programma, nella fase che più di ogni altra permette la qualificazione del prodotto finale.

In un programma di gestione dati infatti, il modo in cui è organizzata la ricerca assume una importanza fondamentale ai fini del lavoro svolto e dei risultati conseguibili durante l'utilizzo del programma stesso. Si è già detto delle difficoltà che possono sorgere nella ricerca di una scheda partendo da un nome: tale situazione è generata da una impostazione che basa la ricerca su una identità, cioè assegnato un dato, lo confronta con tutti i dati di un campo: Se il mio dato è = a un dato del campo Allora Stampa il record corrispondente. In un programma esemplificativo per 5 indirizzi, quanto fin qui detto, diventerebbe:

```
10 FORK = 1TO5
20 INPUT "NOME ";A$(K)
30 INPUT "VIA ";B$(K)
40 INPUT "CITTA ";C$(K)
50 NEXT
60 INPUT "NOME DA CERCARE ";A$
70 FORK = 1TO5
90 IF A$ = A$(K) THEN 130
110 NEXTK
130 ? "NON HO TROVATO";
    A$:GOTO60
140 ?A$(K):?B$(K):?C$(K)
150 GOTO60
```

Per superare l'inconveniente di tale soluzione, cioè per poter effettuare la ricerca con una minima parte del dato posseduto, ad esempio solo il cognome o il nome, si dovrà sviluppare il precedente programma in modo da permettere al computer di effettuare la ricerca non per identità globali (dato = dato), ma per identità parziali dove la presenza del dato da ricercare non è ammessa da un unico confronto, ma da una serie di confronti che permettono di controllare la presenza o meno del nostro dato all'interno del dato posseduto dal computer.

Se chiediamo al programma precedente di cercarci il dato Luigi mentre lui possiede Rossi Luigi, avremo un laconico messaggio "Non ho trovato Luigi".

Per avere una risposta positiva si deve mettere in grado il computer di eseguire una carrellata all'interno del dato da lui posseduto confrontando Luigi con un

numero di caratteri di Rossi Luigi corrispondente alla lunghezza del nostro nome. In pratica il programma dovrebbe compiere questa operazione:

```
(ROSSI) LUIGI No
R(OSSI)LUIGI No
RO(SSIL)UIGI No
ROS(SI LU)IGI No
ROSS(I LUI)GI No
ROSSI(LUIG)I No
ROSSI(LUIGI)SI
```

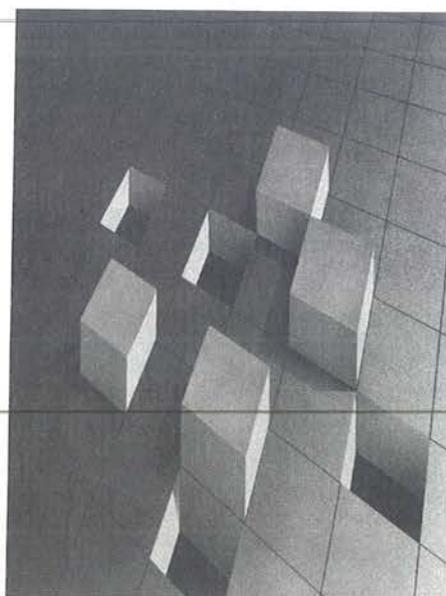
Così presentata può sembrare una cosa lunga, ma per fortuna il BASIC ci permette di ottenere questo risultato con il ricorso ad un solo ciclo For-Next con l'ausilio dell'istruzione Mid\$ per lo spostamento all'interno dei dati posseduti dal computer.

Al listato precedente si dovranno aggiungere o modificare le seguenti linee:

```
80 FORQ = 1TOLEN(A$(K))-LEN(A$)
    + 1
90 IF A$ = MID$(A$(K),Q,LEN(A$))
    THEN N = 1:GOTO130
100 NEXTQ
120 IF N = 1 THEN 60
150 GOTO100
```

La complessità del limite massimo del ciclo di linea 80 serve per non effettuare dei controlli inutili (rimanendo all'esempio: Uigi, lgi, Gi, I). La linea 90 realizza lo spostamento all'interno del dato posseduto dal computer ed esemplificato dalle parentesi, mentre la variabile N serve a controllare il ritrovamento di almeno un blocco dati, nel qual caso la linea 120 impedirà successivamente la stampa del messaggio di mancato ritrovamento della linea 130. Il cambiamento alla linea 150 permette di trovare tutti i dati posseduti dal computer e rispondenti alle caratteristiche da noi richieste: un mancato aggiornamento avrebbe, come effetto, la stampa solo del primo record valido.

Una conseguenza di questa soluzione è che operando secondo tali modalità il computer non troverà solo il record da noi desiderato, ma anche tutti gli altri da lui posseduti e che corrispondono, alle specifiche da noi richieste, per cui, rimandando all'esempio già utilizzato, verranno visualizzati tutti gli indirizzi contenenti nel campo Nome la parola Luigi (quindi anche Pierluigi avrà un riscontro positivo). Se l'elenco di tutti i



## COMMODORE 16

Luigi e affini è il nostro scopo nessun problema, mentre nel caso si vogliono visualizzare solo i dati desiderati si dovrà aggiungere una istruzione che permetta di bloccare la ricerca al ritrovamento di quanto voluto.

Un'ultima considerazione sull'organizzazione di questa routine tanto importante nell'economia di un programma di memorizzazione dati, riguarda la possibilità di scegliere il campo sul quale effettuare la ricerca. Finora questa operazione è stata svolta operando sul campo Nome, ma, anche considerando quanto appena detto sulla possibilità di ricercare interi blocchi di dati con una caratteristica in comune, è comprensibile l'importanza di una opzione, per mezzo della quale poter avere in poco tempo tutti gli indirizzi dei clienti di una data provincia o di quelli che non hanno pagato regolarmente le rate, o tutti i titoli dei libri posseduti suddivisi per argomenti ecc.

Tutto questo comporta naturalmente una serie di operazioni per le quali, dato il loro sviluppo, si rimanda alla parte pratica.

#### Cambiamento

La gestione di questa routine richiede, innanzi tutto, il ricorso alla precedente fase di ricerca e, data l'impostazione appena descritta, si dovrà predisporre una istruzione che permetta di decidere se i dati trovati sono o meno quelli che si desidera cambiare. In caso affermativo si dovranno passare in rassegna tutti i dati per decidere quali devono essere cambiati e quindi apportare le dovute correzioni.

La soluzione più semplice in tal senso è la presentazione di un singolo dato alla

volta con la richiesta di eventuali cambiamenti e, in caso di risposta affermativa, il ricorso a un Input per l'assegnazione del nuovo dato. Lo sviluppo di soluzioni più eleganti è lasciato alla fantasia di ogni programmatore; nel listato presentato in queste pagine verrà proposto il ricorso all'istruzione Get che, con opportuni controlli, permette una gestione più semplice e ordinata dell'operazione descritta.

#### Cancellazione

La prima parte di questa sezione del programma segue fedelmente l'impostazione della precedente con il ricorso alla routine di ricerca dei dati da cancellare. Trovato quanto desiderato si dovranno annullare le variabili in gioco: operazione più semplice rispetto alla fase di cambiamento, ma per ottenere un risultato altrettanto valido si dovrà strutturare il programma in modo che non solo cancelli i dati non necessari, ma recuperi lo spazio lasciato vuoto dagli stessi. Questo permette di non occupare inutilmente dei byte, cosa importante soprattutto quando la memoria a disposizione non è molta, ma per raggiungere questo risultato si dovrà, all'atto della cancellazione, spostare nello spazio del record cancellato i dati dell'ultimo record del file e, a cose fatte, diminuire di 1 la variabile contenente il numero di record utilizzati (figura 2).

È opportuno che, dopo ogni ricorso alle routine di cambiamento o cancellazione, si salvino i dati sulla memoria di massa esterna (cassetta o dischetto) quale salvaguardia dell'efficacia di una operazione che altrimenti perderebbe i suoi effetti allo spegnimento del computer.

#### Registrazione

Indispensabile per il mantenimento dei dati allo spegnimento del computer. Non richiede particolari accorgimenti, a parte l'attenzione che lo sviluppo delle due distinte fasi, di registrazione dei nuovi dati sul supporto magnetico e richiamo dei dati memorizzati ad un successivo utilizzo del programma, sia identico nella concatenazione delle variabili in gioco. Se, per esempio, la prima operazione effettuata nella fase di salvataggio dei dati è la memorizzazione del numero di record utilizzati; ricorrendo alla variabile N, la prima operazione da effettuarsi nella fase di richiamo dei dati memorizzati la volta precedente, sarà il recupero del numero dei record utilizzati ricorrendo, per questo, sempre alla variabile N.

#### La generalizzazione

Un programma per la memorizzazione di dati, sviluppa dei file organizzati in campi e record, ma, mentre la gestione del nome del file e del numero dei record non presenta particolari problemi ed è legato essenzialmente all'aspetto pratico di ogni singola applicazione (il più semplice programma strutturato secondo le direttive sopra discusse è in grado di aprire più file contenenti un numero diverso di record), l'aspetto campi si manifesta più complesso sia in merito al numero degli stessi che all'argomento da essi trattato.

Un programma organizzato per la gestione di indirizzi richiederà un numero e degli argomenti diversi per i suoi campi, rispetto a un programma per la memorizzazione dei dati riguardanti una biblioteca, pur mantenendo una identica strutturazione per quanto riguarda il trattamento dei dati.

Le vie a questo punto sono due: lo sviluppo di più programmi diversi solo nell'organizzazione dell'aspetto campi o lo sviluppo di un solo programma capace di gestire situazioni diverse in merito alla definizione dei campi.

Nel primo caso ogni singolo programma occuperà meno memoria rispetto alla seconda soluzione, ma si dovranno di volta in volta adattare alcune parti del listato in relazione ai bisogni del momento e per passare ad argomenti diversi si dovrà caricare un nuovo pro-

**Figura 2 - Esempio di riorganizzazione del file di figura 1 dopo l'avvenuta cancellazione del precedente record 3. Si può notare il modo di operare del programma durante tale operazione, che consiste nello spostare l'ultimo record al posto di quello cancellato.**

	Campo n.1 NOME	Campo n.2 VIA	Campo n.3 CITTÀ	Campo n.4 TELEFONO
Record 1	Luigi	Via Verdi 4	Milano	67564534
Record 2	Sandro	Via Rossi 7	Milano	23435456
Record 3	Marco	Via Gialli 9	Brescia	6473828
Record 4	Paolo	Via Neri 45	Milano	88765601

# Gli ultimi saranno i primi?

## Giudicalo tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione.

Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

## Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

### CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:  
J. soft - Viale Restelli 5 - 20124 Milano  
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L. ....  
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. .... Cod. ....  
Cod. .... Cod. ....  
Cod. .... Cod. ....

- Contanti allegati  
 Assegno allegato n° .....  
 Ho spedito l'importo a mezzo vaglia postale  
 Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano  
 Pagherò in contrassegno al postino al ricevimento dei volumi

Nome .....

Cognome .....

Via .....

CAP ..... Città ..... Prov. ....

Se richiesta fattura - codice fiscale .....

Data .....

## J. soft: libri di qualità

### Per Commodore 64

**Giochi fantastici per il C 64**  
Cod. ASOC 001 L. 42.000  
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vantaggioso per l'apprendimento delle tecniche di programmazione.

**Il libro del C 64**  
Volume 1  
Cod. ASOC 010 L. 24.000  
(inclusa cassetta)  
**Il libro del C 64**  
Volume 2  
Cod. ASOC 011 L. 24.000  
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.



**C 64: suono e grafica - Volume 1**  
Cod. ASOC 008 L. 24.000  
(inclusa cassetta)  
**C 64: suono e grafica - Volume 2**  
Cod. ASOC 009 L. 24.000  
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.

### Per ZX Spectrum

**15 giochi grafici per ZX Spectrum**  
Cod. ASOC 002 L. 24.000  
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti

all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.



Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.

**Paper Book - programmi per Apple II -**  
Cod. ASOC 005 L. 18.000

**Paper Book - programmi per VIC 20 -**  
Cod. ASOC 003 L. 18.000

**Paper Book - programmi per Commodore 64 -**  
Cod. ASOC 004 L. 18.000

**Paper Book - programmi per Texas TI 99/4A -**  
Cod. ASOC 007 L. 18.000

**Paper Book - programmi per ZX Spectrum -**  
Cod. ASOC 006 L. 18.000



In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, adventure una miscela esplosiva di software "pronto all'uso"!

**J. soft** s.r.l.

Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843

gramma in memoria; nel secondo caso, invece, l'occupazione di un numero maggiore di byte, oltre a permettere l'uso dello stesso programma per la gestione di argomenti diversi, favorisce l'utente inesperto al quale non sarà chiesto di cambiare alcuna parte del listato per formare schedari di differenti argomenti, in quanto per questo gli sarà sufficiente utilizzare correttamente il programma del quale dispone.

### In pratica

Ora vedremo come quanto fin qui detto sia stato messo in pratica nella realizzazione del programma presentato in queste pagine. La costruzione modulare, cioè per compartimenti stagni, permette una comprensione pressoché immediata del listato proposto (si utilizzi in questa fase anche il diagramma a bloc-

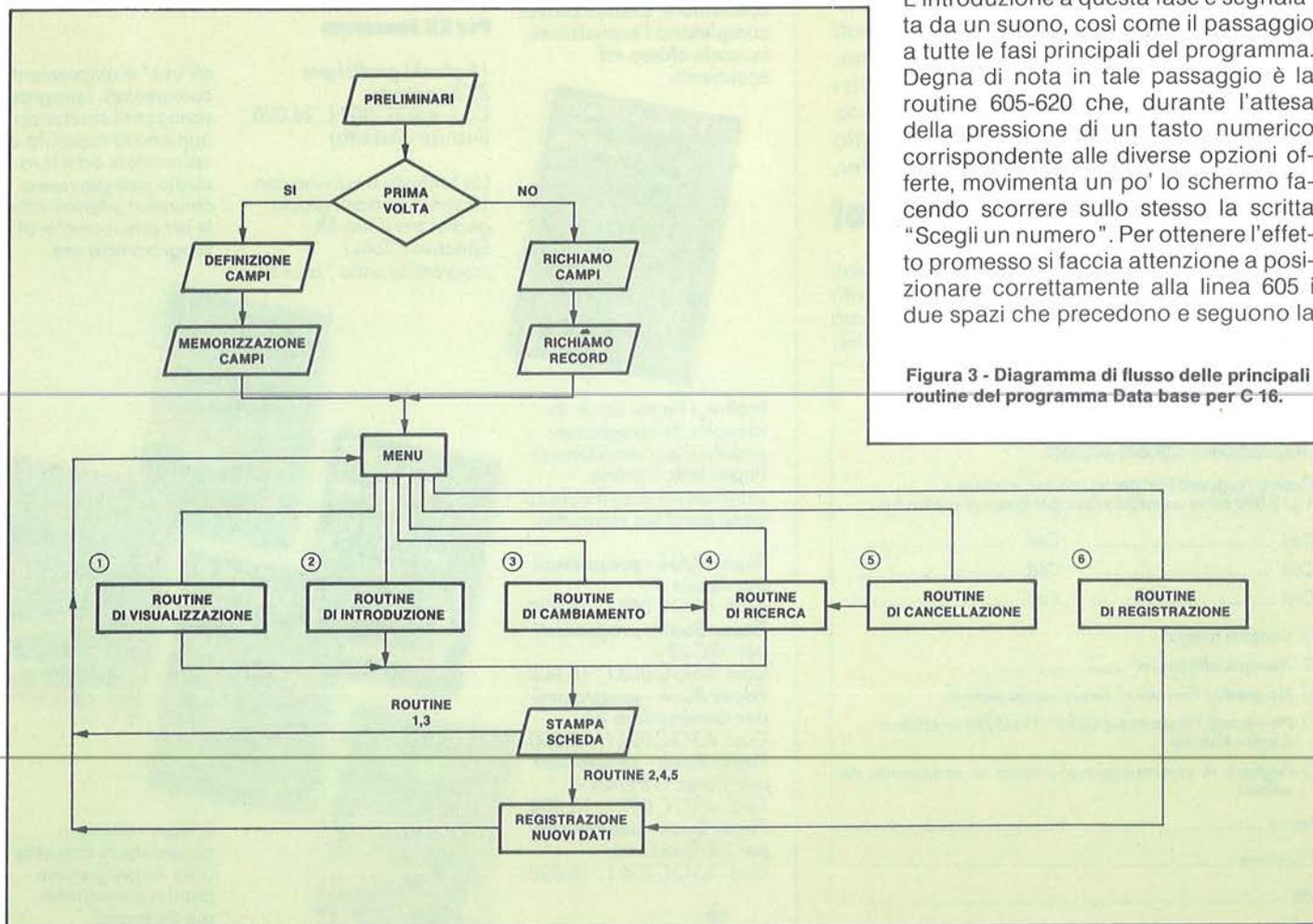
chi relativo alle diverse routine dalla figura 3) oltre a facilitare la gestione di eventuali cambiamenti. Appena dato il Run, dopo il dimensionamento e uno schermo iniziale, (Chr\$(14) fa passare dal maiuscolo al minuscolo), viene chiesto il nome del File (linea 110) e, dopo un controllo dello stesso, abbiamo la prima opzione di una certa rilevanza: alle linee 155-160 si può scegliere se utilizzare le cassette o i dischetti come supporti magnetici.

Alla linea 170 ci si imbatte in una importante scelta effettuata nella strutturazione del programma che è stato organizzato privilegiando la sua generalizzazione. Se la risposta alla linea 170 è affermativa si entra nella routine che permette la definizione del numero e dell'intestazione dei campi con relativo controllo (linee 200-300) per poi passare alla fase di memorizzazione dei dati ap-

pena definiti sul supporto di massa precedentemente scelto (linee 330-380) e quindi andare al menu delle operazioni offerte dal programma.

Se la risposta alla domanda di linea 170 è invece negativa, se cioè sono già stati definiti precedentemente i dati relativi ai campi del file che si intende utilizzare, il programma salta (linea 190) alla linea 390 dove per prima cosa vengono caricate le informazioni proprie dei campi (si noti come le routine 320-370 e 400-450 differiscano solo per il ricorso a Print e Input o allo 0 e 1 utilizzati con Open che definiscono la direzione del movimento dei dati, mentre le variabili utilizzate e il loro ordine siano identici, per quanto detto circa le modalità della fase di Registrazione) e quindi i record introdotti durante gli usi precedenti (notare le linee 470-480 che permettono un corretto uso delle cassette in questa operazione) prima di arrivare al menu. L'introduzione a questa fase è segnalata da un suono, così come il passaggio a tutte le fasi principali del programma. Degna di nota in tale passaggio è la routine 605-620 che, durante l'attesa della pressione di un tasto numerico corrispondente alle diverse opzioni offerte, movimentata un po' lo schermo facendo scorrere sullo stesso la scritta "Scegli un numero". Per ottenere l'effetto promesso si faccia attenzione a posizionare correttamente alla linea 605 i due spazi che precedono e seguono la

Figura 3 - Diagramma di flusso delle principali routine del programma Data base per C-16.



scritta in questione. Per i neofiti il trucco è presto spiegato: si tratta semplicemente di ordinare al computer di spostare il primo carattere della frase in fondo alla stessa (per esempio Ciao, laoc, Aoci, Ocia, ecc.), azione egregiamente svolta dalla sola linea 612, e di stampare la nuova stringa partendo dalla stessa posizione della precedente (linea 610 che sale di una riga, parte dalla stessa colonna e scrive in inverso).

Scelta l'operazione con la quale si intende lavorare, si sente un suono (linea 625 che annulla anche la scrittura in inverso) e si passa all'opportuna routine (linea 630).

## Le routine

### Fine

Le linee da 640 a 690 contengono le istruzioni per terminare il programma

con la possibilità, per chi ne avesse bisogno, di fare una copia dei dati in memoria (è sempre meglio insistere fino alla fine prima di incorrere in qualche dimenticanza fatale).

### Visualizzazione

Si sviluppa dalla linea 700 fino alla 763 con il ricorso alla subroutine 765-790 che permette l'effettiva visualizzazione delle singole schede.

L'utilizzazione offerta è molto semplice e consiste nella stampa della scheda il cui numero, secondo l'ordine di memorizzazione, è influenzabile con la pressione dei tasti <e> con < che fa stampare la scheda con numero inferiore alla precedente e > che permette l'operazione inversa. Si inizia con la prima scheda e si termina (linea 760) quando si chiede la stampa di una scheda dal numero superiore alla quantità di sche-

de memorizzate.

La visualizzazione dei dati è permessa da un semplice ciclo For-Next correato da alcuni segni semigrafici che permettono una miglior presentazione del tutto (linee 765-785).

### Introduzione

Dopo un iniziale, quanto opportuno, controllo della presenza di uno spazio sufficiente alla memorizzazione di nuovi dati (linee 800-807), vengono stampati gli argomenti dei diversi campi con relativa richiesta dei nuovi dati corrispondenti e controllo della loro lunghezza che per problemi di schermo non può superare dei limiti precisi (linee 820-840).

Al termine della scheda c'è un controllo finale della correttezza della stessa (linea 870) e, in caso di risposta affermativa, si deve avvisare il computer sul bi-

Listato 1 - Il programma Data base.

```

10 DIMA$(15):VOL8
20 PRINT"{CLR}{ 5 GIU' }":PRINTCHR$(14)
30 PRINTTAB(14)"[<A>]{ 11 *}[<S>]"
35 PRINTTAB(14)"_{ 11 SPAZI}_"
40 PRINTTAB(14)"_{RVS}DATA BASE{OFF}_"
45 PRINTTAB(14)"_{ 11 SPAZI}_"
50 PRINTTAB(14)"[<Z>]{ 11 *}[<X>]"
60 PRINT"{GIU'}"TAB(17)"PER C16"
70 PRINT"{ 5 GIU'}"TAB(14)"DI CLAUDIO POM
  A"
80 FORT=1TO3000:NEXT
100 PRINT"{CLR}":SOUND1,911,20
110 PRINT"COME SI CHIAMA IL FILE COL QUAL
  E VUOI{ 3 SPAZI}OPERARE "
120 INPUTFI$
130 PRINT:PRINT"VUOI DAVVERO OPERARE CON
  "FI$ " ? ":PRINT"(S/N) ";
140 GETA$:IFA$<>"S"ANDA$<>"N"THEN140
150 IFA$="N"THEN100
155 PRINT:PRINT:PRINT"USI LE CASSETTE O I
  DISCHI ? (C/D) ";
160 GETA$:IFA$<>"C"ANDA$<>"D"THEN160
163 PRINTA$
165 ME=1:IFA$="D"THENME=8
170 PRINT"{ 2 GIU'}E' LA PRIMA VOLTA CHE
  OPERI CON":PRINTFI$(S/N) ";
180 GETA$:IFA$<>"S"ANDA$<>"N"THEN180
190 IFA$="N"THEN390
200 PRINT"{CLR}"
210 PRINT"ALLORA DEVI DEFINIRE I VARI CAM
  PI CHE{ 3 SPAZI}COMPONGONO IL FILE "FI$
  I$
220 PRINT"{GIU'}"PUOI USARNE UN MASSIMO DI
  15."
```

```

225 PRINT"SE NON TI SERVONO TUTTI PUOI PR
  EMERE E{ 3 SPAZI}PER FINIRE.{GIU'}"
230 FORT=1TO15:PRINT"CAMPO N."T "-> ";
240 INPUTA$(T):IFA$(T)="E"ORA$(T)="E"THEN
  260
250 NEXT
260 PRINT"{CLR}{RVS}QUESTI{SPAZI}SONO
  {SPAZI}I{SPAZI}CAMPI{SPAZI}CHE{SPAZI}
  HAI{SPAZI}SCELTO{OFF}{GIU'}":NC=T-1
270 FORT=1TONC:PRINTA$(T):NEXT
280 PRINT"{ 2 GIU'}"VA TUTTO BENE ? (S/N)
  ";
290 GETA$:IFA$<>"N"ANDA$<>"S"THEN290
300 IFA$="N"THEN100
310 GOSUB940
320 OPEN1,ME,1,"MEM"+FI$
330 PRINT#1,NC
340 FORT=1TONC
350 PRINT#1,A$(T)
360 NEXT
370 CLOSE1
380 DIMRE$(NC,50):GOTO500
390 GOSUB940
400 OPEN1,ME,0,"MEM"+FI$
410 INPUT#1,NC
420 FORT=1TONC
430 INPUT#1,A$(T)
440 NEXT
450 CLOSE1
460 IFME=8THEN490
470 PRINT:PRINT"{RVS}E ORA LA CASSETTA CO
  N I DATI{OFF}"
480 PRINT:PRINT"PREMI RETURN PER CONTINUA
  RE ";:INPUTA$
490 DIMRE$(NC,50):GOSUB1105
500 PRINT"{CLR}":PRINTCHR$(142)
```



## COMMODORE 16

sogno o meno di inserire altri dati. A questo punto premendo S si ripete la fase di immissione utilizzando un numero aggiornato per la nuova scheda, mentre alla pressione di N si passa al salvataggio dei dati sulla memoria di massa esterna prescelta (dischetto o cassetta) e quindi si ritorna al menu (linea 930).

#### Ricerca

Questa operazione può essere utilizzata sia da sola per ottenere un dato che ci interessa, oppure all'interno delle fasi per il cambiamento o la cancellazione dei dati. Per permettere questo uso allargato si è dovuto trattare tale fase come una subroutine alla quale accedere con il ricorso a Gosub e questo spiega la linea 1200 che potrebbe altrimenti essere ritenuta inutile. Non si tratta naturalmente dell'unica via possibile: si può ottenere lo stesso effetto tralasciando la linea 1200 e aggiungendo un controllo del Return alla linea 1400, ma si occuperebbe qualche byte di più.

Alla linea 1220 vengono stampati, numerati, i titoli dei campi del file col quale si sta lavorando e quindi si passa alla richiesta del numero del campo col quale si vuole effettuare la ricerca. Controllato che il valore scelto ricada nell'intervallo corretto (linea 1250) si deve introdurre l'insieme di caratteri che si vuole cercare. In relazione alla costruzione del programma, le cui modalità sono state spiegate precedentemente, non è necessario dare al computer il dato nel modo esatto con cui è memorizzato, ma basterà proporre una piccola parte con l'ovvio inconveniente che così facendo la ricerca potrà avere più esiti positivi, mentre il vantaggio di tale organizzazione consiste nella possibilità di poter estrarre dalla massa di dati tutti quelli rispondenti al parametro scelto.

Avendo inizialmente predisposto la memorizzazione dei dati in una variabile con indice a due dimensioni, è ora possibile sintetizzare tutta la fase della ricerca in due semplici cicli For-Next (linea 1280-1320), che al di là del nome delle variabili utilizzate, sono uguali a quelli sviluppati per il programmino esemplificativo durante l'approccio teorico (linee 70-110). Il ricorso a una matrice permette in più, rispetto all'esempio, di effettuare, con le stesse istruzioni, la ricerca su parametri diversi e semplifica la succes-

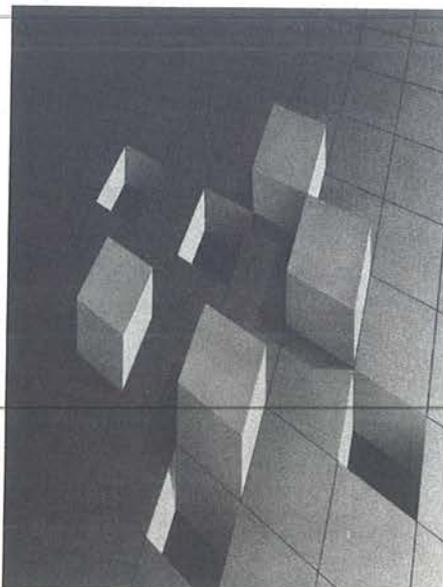
siva fase di stampa della scheda per la quale, onde risparmiare in memoria, si ricorre (linea 1350) alla routine già analizzata durante la descrizione della Visualizzazione.

Quando la ricerca ha esito positivo, dopo la stampa della relativa scheda viene chiesto se si tratta dei dati desiderati, per avviare a quanto detto sulla possibilità di avere più di un ritrovamento per la particolare procedura adottata per la ricerca. Una risposta negativa fa continuare l'operazione sui dati della scheda seguente (linea 1320), mentre un sì rimanda alla routine di partenza (linea 1400) che può essere quella di Cambiamento, di Cancellazione oppure una Ricerca fine a sé stessa, nel qual caso (linea 1200) si torna direttamente al menu.

Lo stesso avviene quando il programma ha controllato anche l'ultima scheda, con l'aggiunta che, in tale situazione, viene stampato un messaggio che informa circa il termine della ricerca per mancanza di ulteriori dati (linea 1330).

#### Cambiamento

La prima operazione eseguita è la ricerca della scheda nella quale si vogliono apportare dei cambiamenti. Se la ricerca ha esito negativo, la variabile H (linea 1340 e linea 1410) rimanda immediatamente al menu, mentre in caso affermativo viene eseguito il ciclo compreso tra le linee 1430 e 1520 la cui caratteristica è il ricorso all'istruzione Get.



Questa soluzione presenta il vantaggio di non richiedere l'uso per ogni singolo dato di più Input per la richiesta dell'opportunità di cambiare il dato stesso e per l'eventuale sostituzione. Infatti la linea 1450 scrive il dato in questione, non ci si dimentichi il punto e virgola (;) senza il quale il risultato sarebbe ben diverso, e il Get di linea 1460 ne permette l'accettazione o la correzione immediata e molto pratica grazie ai controlli delle linee successive:

- la linea 1470 (13 in codice ASCII corrisponde alla pressione del tasto Return) simulando un Input indica l'avvenuto controllo, ed eventuale cambiamento effettuato, sul dato in questione e ci manda al dato seguente;

- le linee 1480 e 1490 permettono la cancellazione di un singolo carattere (20=DEL);

- la linea 1495 controlla la lunghezza del dato e qualora raggiunga il limite massimo non permette ulteriori aggiunte;

- la linea 1500 controlla il tasto premuto e permette l'inserimento solo di determinati caratteri (lettere dell'alfabeto, numeri, simboli di punteggiatura e matematici) al fine di salvaguardare l'utente inesperto dalle pressioni di particolari tasti che potrebbero creare degli inconvenienti al corretto funzionamento del programma.

Terminata la correzione viene stampata la scheda aggiornata (linea 1530) e, dopo l'assenso, i nuovi dati vengono salvati sulla memoria di massa (linea 1580) per poi tornare al menu.

#### Cancellazione

L'inizio è identico alla fase di cambiamento (si confrontino le linee 1410 e 1600). All'esito positivo della ricerca viene offerta un'ultima possibilità di uscire dall'operazione e di non cancellare i dati che, nel caso non si cambi idea, vengono irrimediabilmente persi, in quanto al loro posto viene memorizzata l'ultima scheda del file (linee 1640-1660) mentre il numero dei record viene ridotto di un punto e la nuova situazione è sancita dal ricorso alla registrazione magnetica.

#### Registrazione

Le due distinte routine per la scrittura su dischetto o cassetta e la lettura dagli stessi, non hanno bisogno di particolari

Seguito listato Data base.

```

505 SOUND1,917,20
510 PRINTTAB(12){ 2 SU}U{ 15 *}I"
520 PRINTTAB(12)"_{RVS}MĒNU' DĀTĀ BASE
{OFF}-"
530 PRINTTAB(12)"J{ 15 *}K"
540 PRINT"{ 3 GIU}"TAB(7)"1){ 2 SPAZI}VI
SUALIZZAZIONE DATI"
550 PRINT"{GIU}"TAB(7)"2){ 2 SPAZI}INTRO
DUZIONE NUOVI DATI"
560 PRINT"{GIU}"TAB(7)"3){ 2 SPAZI}RICER
CA DATI"
570 PRINT"{GIU}"TAB(7)"4){ 2 SPAZI}CAMBI
AMENTO DATI"
580 PRINT"{GIU}"TAB(7)"5){ 2 SPAZI}CANCE
LLAZIONE DATI"
590 PRINT"{GIU}"TAB(7)"6){ 2 SPAZI}REGIS
TRAZIONE DATI"
600 PRINT"{GIU}"TAB(7)"7){ 2 SPAZI}FINE
{ 4 GIU}"
605 B$="{ 2 SPAZI}SCEGLI UN NUMERO
{ 2 SPAZI}"
610 PRINTTAB(10){SU}{RVS}"B$
611 FORF=1TO200:NEXT
612 B$=MID$(B$,2,19)+LEFT$(B$,1)
620 GETA$:N=VAL(A$):IFN<1ORN>7THEN610
625 PRINT"{OFF}":SOUND1,864,15
630 ONNGOTO700,800,1200,1410,1600,1680,64
0
640 PRINT"{CLR}"
650 PRINT"VUOI FARE UNA COPIA DEI NUOVI D
ATI ?{ 4 SPAZI}(S/N) ";
660 GETA$:IFA$="N"THEN690
670 IFA$<"S"THEN660
680 GOSUB1000
690 PRINT"{CLR}{ 12 GIU}"TAB(18)"CIAO.":
END
700 N=0:PRINT"{CLR}":PRINTCHR$(14)
710 PRINT"{HOME}{ 18 GIU}"
715 PRINT"<A>{ 38 *}[<S>";
720 PRINT"- - > -{ 2 SPAZI}AVANTI
{ 9 SPAZI}- < - INDIETRO -";
725 PRINT"<Z>{ 38 *}[<X>";
730 GETA$:IFA$<"."AND$<"."THEN730
740 IFA$="."ANDN<=NUTHENN=N+1
750 IFA$="."ANDN>1THENN=N-1
760 IFN>NUTHEN790
763 GOSUB765:GOTO710
765 PRINT"{CLR}[<A>{ 38 *}[<S>";
770 PRINT"-";SPC(15){RVS}SCHEDA N.{OFF}
"N";PRINTSPC(11-LEN((STR$(N)+STR$(NU)
))NU);
775 PRINT"-";PRINT"<Q>{ 38 *}[<W>";
780 FORT=1TONC:PRINT"-{DES}"RE$(T,N):PRIN
T"{SIN}-";NEXT
785 PRINT"<Z>{ 38 *}[<X>":RETURN
790 PRINT"{GIU'}{RVS}RICERCA{SPAZI}TERMI
NATA{OFF} PREMI <RETURN> ";:INPUTA$:G
OTO500
800 IFNC*30<FRE(I)ANDNU<50THENNU=NU+1:PRI
NTCHR$(14):GOTO810
805 PRINT"{CLR}{GIU'}NON C'E' PIU' SPAZIO
IN MEMORIA{ 2 GIU'}"
806 INPUT"PREMI IL TASTO RETURN ";W$:GOTO
500
810 PRINT"{CLR}"TAB(16){RVS}SCHEDA
{SPAZI}N.{OFF} "NU:PRINT
820 FORT=1TONC
830 PRINTA$(T);:INPUT"-> ";RE$(T,NU)
835 IFLEN(RE$(T,NU))>36THENPRINT:PRINT"TR
OPPO LUNGO ! MAX.36 ":PRINT:GOTO830
836 IFLEN(RE$(T,NU))=0THENRE$(T,NU)="-"
840 NEXT
850 S=N:N=NU:GOSUB765:N=S
870 PRINT"{ 2 GIU'}VA TUTTO BENE ? (S/N)
";
880 GETA$:IFA$="N"THEN810
890 IFA$<"S"THEN880
900 PRINT:PRINT"{ 2 GIU'}ANCORA ? (S/N) "
;
910 GETA$:IFA$="S"THEN800
920 IFA$<"N"THEN910
930 GOSUB1000:GOTO500
940 PRINTCHR$(14){CLR}"
950 A$="LA CASSETTA":IFME=8THENA$="IL DIS
CHETTO"
960 PRINT"{GIU'}HAI PREDISPOSTO "A$ ? (
S/N) ";
970 GETA$:IFA$="S"THENRETURN
980 IFA$<"N"THEN970
990 PRINT:PRINT"{GIU'}E COSA ASPETTI A FA
RLO ?":FORT=1TO3000:NEXT:GOTO950
999 REM *** SCRITTURA SU DISCO O CASSETTA
***
1000 GOSUB940
1010 IFME=8THENOPEN1,ME,15,"S0:"+FI$:CLOS
E1
1015 OPEN1,ME,1,""+FI$
1020 PRINT#1,NU
1030 FORT=1TONU
1040 FORT=1TONC
1050 PRINT#1,RE$(T1,T)
1060 NEXT
1070 NEXT
1080 CLOSE1
1090 RETURN
1099 REM *** LETTURA DA DISCO ***
1100 GOSUB940
1105 REM
1110 OPEN1,ME,0,""+FI$
1120 INPUT#1,NU
1130 FORT=1TONU
1140 FORT=1TONC
1150 INPUT#1,RE$(T1,T)
1160 NEXTT1
1170 NEXTT
1180 CLOSE1
1190 RETURN
1200 GOSUB1210:GOTO500
1210 PRINTCHR$(14){CLR}"
1220 FORT=1TONC:PRINTT" "A$(T):NEXT
1230 PRINT"{ 2 GIU'}DIMMI IL NUMERO DEL P

```

## COMMODORE 16

spiegazioni. Si presti attenzione al punto già accennato e relativo al bisogno che le due fasi siano strutturate in modo identico, con la stessa successione delle stesse variabili. L'unica differenza è la presenza della linea 1010 nell'operazione di scrittura: tale procedura è importante nel caso si usino i dischetti, in quanto salvaguarda da eventuali problemi che possono sorgere qualora venga registrato un file che, per delle cancellazioni, fosse più corto del precedente. La linea in questione non fa altro che annullare il vecchio file prima della memorizzazione del nuovo.

Se i listati non presentano difficoltà particolari, sono invece importanti alcuni consigli per chi ricorrerà alle cassette per il salvataggio dei dati.

Questo supporto non è controllato da un DOS, per cui, nonostante il programma cerchi di dare una mano dove gli è

possibile, la correttezza dell'esecuzione di tutte le operazioni è lasciata alla capacità del singolo utente. Per riuscire nell'intento senza correre il rischio di dannarsi, per interi file andati persi, si devono osservare queste regole:

- utilizzare una cassetta per ogni file;
- di ogni cassetta usare una facciata per la memorizzazione dei titoli dei campi e l'altra per i record;
- segnare sulla cassetta il nome del file e i lati campi e record;
- alla fine di ogni utilizzo di un file fare una copia dello stesso (anche successivamente alla prima registrazione sul lato record).

### Considerazioni finali

Il programma presentato, offre delle prestazioni non indifferenti, dato che il modello di computer sul quale è imple-

mentato, non è certo esente da critiche. Di seguito le motivazioni circa le scelte effettuate e le possibili alternative.

I punti "deboli" sono: il mancato ricorso alla stampante, l'impossibilità di effettuare una ricerca su più campi, il ricorso alla memorizzazione su cassette e ai file sequenziali e non relativi.

I primi due sono riconducibili allo stesso problema: data la limitata quantità di memoria disponibile sul C 16 sono state necessarie delle scelte a vantaggio di certe soluzioni e a danno di altre; in quanto svilupparle tutte avrebbe portato ad un perentorio "Out of memory". Ricordiamoci che la memoria disponibile per la programmazione in BASIC non può essere occupata totalmente dal listato, ma che uno spazio sufficiente deve essere lasciato per le variabili utilizzate e, nel caso di variabili con indice, oltre ai byte occupati al dimensionamento, si

#### Seguito listato Data base.

```

ARAMETRO ";
1240 GETA$:A=VAL(A$)
1250 IFA<LORA>NCTHEN1240
1255 PRINTA
1260 PRINT:PRINT"{GIU'}DIMMI COSA DEVO CE
RCARE E PREMI <RETURN>"
1270 INPUTB$:B=LEN(B$)
1280 FORK=1TONU
1290 FORR=1TOLEN(RE$(A,K))+1-B
1300 IFMID$(RE$(A,K),R,B)=B$THEN1350
1310 NEXTR
1320 NEXTK
1330 PRINT:PRINT"{GIU'}{RVS}RICERCA
{SPAZI}TERMINATA{OFF} PREMI <RETURN
> ":SOUND1,810,15
1340 INPUTA$:H=1:RETURN
1350 S=N:N=K:GOSUB765:N=S
1370 PRINT"{GIU'}E'{SPAZI}QUESTA{SPAZI}L
A{SPAZI}SCHEDE{SPAZI}CHE{SPAZI}VOLE
VI ? (S/N) ";:SOUND1,810,15
1380 GETK$:IFK$="N"THEN1320
1390 IFK$<>"S"THEN1380
1400 PRINT:PRINT"{GIU'}PREMI <RETURN> PER
CONTINUARE ";:INPUTA$:RETURN
1410 GOSUB1210:IFH=1THENH=0:GOTO500
1420 PRINT"{CLR}"
1430 FORT=1TONC
1440 PRINT"{HOME}":FORT=1TOT:PRINT"
{GIU'}"+NEXTT1
1450 PRINTRE$(T,K) " ";
1460 GETB$:IFB$=""THEN1460
1465 B=ASC(B$)
1470 IFB=13THEN1520
1480 IFB=20ANDLEN(RE$(T,K))=1THENRE$(T,K)

```

```

="":GOTO1510
1490 IFB=20ANDLEN(RE$(T,K))>1THENRE$(T,K)
=LEFT$(RE$(T,K),LEN(RE$(T,K))-1):GOT
O1510
1495 IFLEN(RE$(T,K))>35THEN1510
1500 IF(B>31ANDB<91)OR(B>159ANDB<219)THEN
RE$(T,K)=RE$(T,K)+CHR$(B)
1510 GOTO1440
1520 NEXTT
1530 S=N:N=K:GOSUB765:N=S
1550 PRINT"{ 2 GIU'}TUTTO BENE ? (S/N) ";
1560 GETA$:IFAS="N"THEN1420
1570 IFAS<>"S"THEN1560
1580 GOSUB1000:GOTO500
1600 GOSUB1210:IFH=1THENH=0:GOTO500
1610 PRINT:PRINTTAB(3)"VUOI PROPRIO CANC
E{ 2 L}ARLA ? (S/N) ";
1620 GETA$:IFAS="N"THEN500
1630 IFAS<>"S"THEN1620
1640 FORT=1TONC
1650 RE$(T,K)=RE$(T,NU)
1660 NEXT
1670 NU=NU-1:GOSUB1000:GOTO500
1680 GOSUB1000:GOTO500

```

## Elenco delle principali routine

<b>10-80</b>	Schermo introduttivo.
<b>100-165</b>	Domande iniziali.
<b>170-495</b>	Routine di Generalizzazione.
200-300	Inizializzazione numero e titolo dei campi.
310-380	Scrittura dei nuovi dati sulla memoria di massa.
390-490	Lettura dei dati riguardanti campi e record definiti durante un utilizzo precedente.
<b>500-630</b>	Menu.
<b>640-690</b>	Routine di Fine.
<b>700-763</b>	Routine di Visualizzazione delle schede.
<b>765-790</b>	Subroutine di stampa delle schede.
<b>800-930</b>	Routine di Introduzione nuovi dati.
800-807	Controllo memoria disponibile
810-840	Introduzione nuovi dati
850-920	Controllo validità dati ed eventuale nuova immissione.
930	Registrazione e ritorno al menu.
<b>940-990</b>	Subroutine per il controllo della predisposizione della opportuna memoria di massa.
<b>1000-1090</b>	Routine di scrittura su disco o cassetta.
<b>1100-1190</b>	Routine di lettura da disco o cassetta.
<b>1200-1400</b>	Routine di Ricerca.
1220	Stampa campi.
1230-1255	Scelta campo.
1260-1270	Scelta parola da cercare.
1280-1320	Fase di ricerca.
1330-1340	Ricerca terminata.
1350-1390	Controllo validità scheda trovata.
<b>1410-1580</b>	Routine di Cambiamento.
1410	Salto alla subroutine di Ricerca.
1430-1520	Ciclo di presentazione e correzione dei dati.
1530-1570	Stampa e controllo nuova scheda.
1580	Registrazione nuovi dati e ritorno al menu.
<b>1600-1670</b>	Routine di Cancellazione.
1600	Salto alla subroutine di Ricerca.
1610-1630	Controllo del ricorso alla cancellazione.
1640-1660	Passaggio dell'ultima scheda al posto di quella da cancellare.
1670	Aggiornamento numero record, registrazione nuovi dati e ritorno al menu.
<b>1680</b>	Routine di Registrazione per fare una copia del file.

## Elenco delle principali variabili

<b>RES(x,y)</b>	Matrice per la memorizzazione dei record.
<b>AS(x)</b>	Contiene i titoli dei campi.
<b>FIS</b>	Nome del file.
<b>NC</b>	Numero dei campi del file.
<b>NU</b>	Numero dei record utilizzati.
<b>ME</b>	Serve per distinguere la memoria di massa utilizzata (cassette o dischetti).
<b>N</b>	Numero della scheda sulla quale si sta operando.

devono prevedere, e quindi lasciare liberi, quelli che serviranno all'introduzione successiva dei vari dati: 50 record con 5 campi occupati in media 20 byte, richiedono ben 5.000 byte per la memorizzazione dei dati nelle variabili relative, e se non ci sono, incominciano i guai, a meno che non si sia corso ai ripari con linee come la 800.

In considerazione di tali problemi, durante la cernita delle operazioni da inserire nel programma si è tentato di privilegiare l'utente medio del C 16, preferendo routine che un "neocommodoriano" difficilmente è in grado di sviluppare da solo e tralasciando quelle non indispensabili alla probabile configurazione tipo di un acquirente del C 16, che nella maggioranza dei casi non possiede una stampante, e all'utilizzo che questi farà del programma, senz'altro tale da non richiedere sofisticate procedure di ricerca.

Per quanto riguarda la possibilità di memorizzare i file su cassetta si è tenuto conto delle esigenze della maggioranza, pur avvisando circa le difficoltà di tale procedura; mentre il ricorso a file relativi non è stato sviluppato in quanto il permettere l'uso delle cassette, che non sono in grado di gestire tale tipo di file, richiede una routine a parte per questa soluzione, con un aggravamento dei già citati problemi di memoria. A questo punto è d'obbligo una difesa della scelta di generalizzare il programma: operazione oltremodo dispendiosa per la memoria e facilmente sintetizzabile con un Read e pochi Data.

La soluzione proposta è sorta dal bisogno di offrire qualcosa di valido, pur nell'ambito delle scelte obbligate dalla struttura della macchina, non solo sotto l'aspetto del risultato finale, ma anche dal punto di vista didattico, per cui sono stati risolti i problemi più complessi, ma nello stesso tempo utili anche a chi possiede la configurazione minima. Fra questi rientra senz'altro di più la generalizzazione del programma, che è uno dei punti "deboli" accennati che, tra l'altro, data la progettazione modulare del programma, è facilmente inseribile al posto di routine considerate personalmente inutili.

In questo modo i possessori di dischetti potranno trasformare le operazioni di registrazione ricorrendo ai file relativi; chi ha una stampante può inserire una

## COMMODORE 16

minima serie di istruzioni in grado di soddisfare i propri desideri e chi volesse avere più memoria da mettere a disposizione dei file potrà annullare la fase di generalizzazione.

In aiuto di chi sceglie questa soluzione, si dovranno annullare le linee comprese tra 210-380 e 400-480, modificare la 200 in:

200 GOTO495

e aggiungere:

11 FORT = 1TO15

12 READA\$(T)

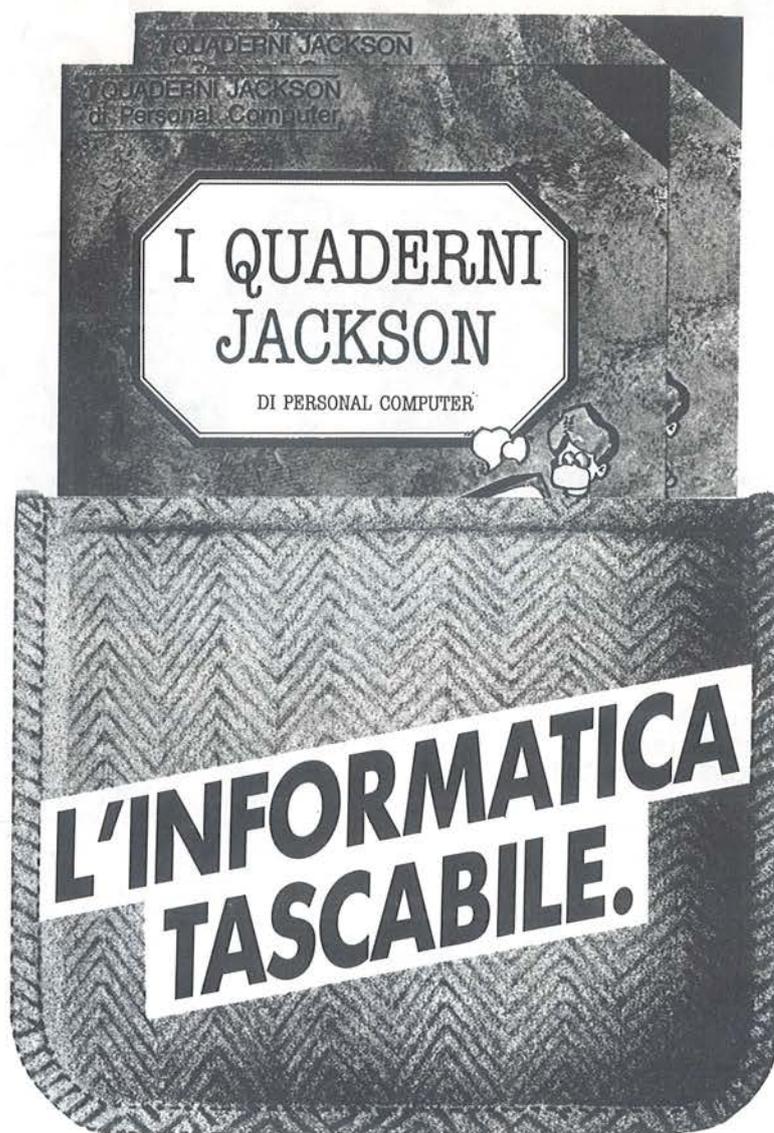
13 IFA\$(T) = "★" THEN NC = T-1

:GOTO20

14 NEXT

e una serie di Data terminanti con ★, per esempio 15 Data Nome e Cognome, Via e Numero, CAP Città PR, Telefono, ★ che ci permetterà di operare con 4 campi e di memorizzare degli indirizzi. Con un numero diverso di campi sarà sufficiente modificare opportunamente la linea dei Data purché si faccia attenzione a terminarla con ★ che serve per il controllo di linea 13.

Ricordando che il programma gira perfettamente anche sugli altri modelli Commodore, che avendo più memoria a disposizione possono accettare qualunque ampliamento senza problemi (per il C 64 si dovranno annullare solo le istruzioni per il sonoro, iniziati sempre con Sound), un ultimo consiglio per risparmiare più memoria possibile: scrivete tutte le istruzioni ricorrendo alle abbreviazioni permesse. I più "esperti" potranno anche diminuire il numero di linee utilizzate concatenando più istruzioni sulla stessa linea (operazione non sempre sviluppata per problemi di chiarezza del listato), ma attenzione ai vari Goto, Gosub e If Then che spesso a tale manovra rispondo con dei messaggi di errore o con non corretti funzionamenti del programma se non viene mantenuto lo sviluppo logico dello stesso. ■



Arrivano i Quaderni Jackson, tanti volumi monografici per conoscere bene il personal computer e l'informatica.

Nei quaderni Jackson c'è tutto quello che è importante sapere sui computer, la programmazione, i linguaggi, il software, le applicazioni e i nuovi sviluppi dell'informatica.

Quaderni Jackson: l'informatica a tutti i livelli, in una collana aperta, pratica, essenziale, aggiornata.

L'informatica tascabile per chi vuole saperne di più e compiere così un salto di qualità nel mondo di oggi e di domani.

Ogni mese, 2 volumi.

### Volumi già pubblicati:

Gianni Giaccaglini

"Vivere col Personal Computer"

Paolo Bozzola

"Dentro e fuori la scatola"

Enrico Odetti

"Ed è subito BASIC Vol. I"

"Ed è subito BASIC Vol. II"

Paolo Capobussi

e Marco Giacobazzi

"A ciascuno il suo Personal"

Fulvio Francesconi

e Fernando Paterlini

"To do or not to do"

In edicola,  
a sole lire 6.000.



**GRUPPO  
EDITORIALE  
JACKSON**

SAN FRANCISCO - LONDRA - MILANO



**Introduzione**

**S**copo del programma proposto, è quello di facilitare la gestione delle videate che implicano operazioni di introduzione di dati da tastiera.

È infatti quasi sempre presente nell'utente, l'esigenza di disporre di una procedura di immissione dati che effettui un minimo di controllo sui tasti battuti.

Se il test che normalmente si effettua alla fine di una videata non fornisce una risposta positiva, è inoltre necessario cancellare velocemente dallo schermo le informazioni da sostituire - lasciando inalterate quelle corrette - e ricominciare la procedura di immissione.

Il programma Easy Video aggiunge alcune routine in linguaggio macchina, richiamabili tramite il comando BASIC Sys, che facilitano le operazioni di input, il posizionamento del cursore e la cancellazione immediata di parti definite dello schermo.

**Il programma**

Una volta caricato e lanciato, il programma legge in circa 10 secondi le routine specificate sotto forma di frasi Data, e le scrive collocandole nell'area di memoria \$C000 ÷ \$C31A (decimale 49152 ÷ 49946).

Dopo un controllo sull'esattezza delle frasi Data, il programma visualizza l'elenco delle 6 routine disponibili e gli indirizzi (in decimale) per il loro richiamo. Quindi elenca le istruzioni necessarie per la loro utilizzazione. Infine si autocancella, liberando l'area BASIC, che resta completamente a disposizione dell'utente.

In figura 1 è riportato un fac-simile delle informazioni che compaiono sul video dopo il lancio del programma.

Qui di seguito viene descritto il funzionamento delle singole routine, mentre il disassemblato commentato del listato 1 dovrebbe soddisfare l'esigenza di chi intende apportare delle modifiche.

**Le routine**

★ *SYSCRIPT*, <char.num.> ["prompt"]; <var.>

È la routine più importante, che ha come scopo quello di facilitare la gestione della immissione dei dati da tastiera.

# Easy Video 64

## Gestione degli input da tastiera

di Maurizio Paolinelli

Essa stampa, a partire dalla posizione in cui il cursore si trova al momento del suo richiamo, l'eventuale messaggio prompt (facoltativo) seguito da un numero di spazi in reverse pari al valore specificato da <char.num.>.

<char.num.> specifica in definitiva il campo entro cui dovrà essere immessa l'informazione.

Viene quindi generato un "finto" cursore lampeggiante, posizionato all'inizio del campo di immissione. L'aspetto del cursore è funzione del tipo di variabile <var.> che viene specificata per ricevere il dato in ingresso: i simboli \$,% o . avvertono che la variabile <var.> è stata definita rispettivamente come variabile stringa, numerica intera o numerica reale.

Da questo momento si attende l'introduzione di un dato da tastiera che non potrà avere un numero di caratteri superiore alla lunghezza del campo evidenziato in reverse (char.num.).

Battendo un tasto, comparirà l'eco - in reverse - del carattere scelto nella posi-

zione indicata dal falso cursore lampeggiante, e quest'ultimo si sposterà di una colonna verso destra segnalando l'attesa di un eventuale nuovo carattere.

Il processo continua fino a quando non venga premuto Return. In tal caso, il contenuto del campo viene memorizzato nella variabile <var.>, il campo in reverse viene sostituito dall'eco - in normal - del dato immesso e il controllo viene restituito al programma che aveva chiamato la routine Cipt.

Se la variabile <var.> è di tipo numerico e il dato introdotto è alfanumerico, viene generato un impulso sonoro, visualizzato per circa un secondo il messaggio "? Redo" e quindi di nuovo il campo vuoto in reverse col cursore pronto per una nuova immissione di dati.

Se si tenta di introdurre un numero di caratteri superiore a quello definito dal campo in reverse (<char.num.>) viene generato un impulso sonoro, cancellato il contenuto del campo in reverse e riportato il cursore all'inizio del campo

```
*****
*** easy video per c.64 ***
*****
```

Sintassi dei nuovi comandi

```
CIPT=49175          EL=49859
PLOT=49852          EF=49881
BIP=48780           CL=49895
```

```
SYSCRIPT,<char.num.>["prompt"];<var.>
SYSLOT,<column>,<row>
SYBIP
SYSEL,<column>,<row>
SYSEF,<column>,<row>
SYSCL,<col.1>,<row1>,<col.2>,<row2>
```

Figura 1 - I nuovi comandi della routine Easy Video.

### Listato 1 - Il disassemblato commentato della routine.

\$C000 #500 Registro memorizzazione numero caratteri campo (NC);  
 \$C001 #500 registro memorizzazione colonna iniziale cursore;  
 \$C002 #500 registro memorizzazione riga iniziale cursore;  
 \$C003 #500 registro memorizzazione n° caratteri inseriti;  
 \$C004 #500 counter per timer;  
 \$C005 #500 registro memorizzazione colonna corrente cursore;  
 \$C006 #500 registro memorizzazione riga corrente cursore;  
 \$C007 #500 registro memorizzazione 'flag' cursore;  
 \$C008 #512 .BY #512 (RVS ON);  
 \$C009 #520 .BY #520 (Spave);  
 \$C00A #592 .BY #592 (RVS OFF);  
 \$C00B #590 .BY #590 (CRSR LEFT);  
 \$C00C #520 .BY #520 (Space);  
 \$C00D #590 .BY #590 (CRSR LEFT);  
 \$C00E #53F .BY #53F, 'R', 'E', 'D', 'O', #500;  
 \$C0014 #500 registro memorizzazione 'flag' scrolling;  
 \$C0015 #500 registro memorizzazione NC+Y.  
 \$C0016 NOP  
 \$C017 JSR \$B7F1  
 \$C001A CPX #54F  
 \$C001C BCC \$C0021  
 \$C001E JMP \$B248  
 \$C0021 STX \$C000  
 \$C0024 JSR \$0079  
 \$C0027 CMP #522  
 \$C0029 BEQ \$C0033  
 \$C002B LDA #538  
 \$C002D JSR \$AEFF  
 \$C0030 JMP \$C003E  
 \$C0033 JSR \$AEB0  
 \$C0036 LDA #538  
 \$C0038 JSR \$AEFF  
 \$C003E JSR \$AB21  
 \$C003E JSR \$B3A6  
 \$C0041 LDA #52C  
 \$C0043 STA \$01EF  
 \$C0046 JSR \$B08B  
 \$C0049 STA \$49  
 \$C004B STY \$4A  
 \$C004D SEC  
 \$C004E JSR \$FFFO  
 \$C0051 STX \$C002  
 \$C0054 STX \$C006  
 \$C0057 STY \$C001  
 \$C005A STY \$C005  
 \$C005D LDA #500  
 \$C005F STA \$C014  
 \$C0062 CLC  
 \$C0063 LDA \$C000  
 \$C0066 ADC \$C001  
 \$C0069 STA \$C015  
 \$C006C CMP #54F  
 \$C006E BCC \$C07C  
 \$C0070 SEC  
 \$C0071 LDA #54E  
 \$C0073 STA \$C015  
 \$C0076 SBC \$C001  
 \$C0079 STA \$C000  
 \$C007C CPX #518  
 \$C007E BCC \$C091  
 \$C0080 LDA \$C015  
 \$C0083 CMP #527  
 \$C0085 BCC \$C091  
 \$C0087 INC \$C014  
 Registro memorizzazione numero caratteri campo (NC);  
 registro memorizzazione colonna iniziale cursore;  
 registro memorizzazione riga iniziale cursore;  
 counter per timer;  
 registro memorizzazione colonna corrente cursore;  
 registro memorizzazione riga corrente cursore;  
 registro memorizzazione 'flag' cursore;  
 .BY #512 (RVS ON);  
 .BY #520 (Spave);  
 .BY #592 (RVS OFF);  
 .BY #590 (CRSR LEFT);  
 .BY #520 (Space);  
 .BY #590 (CRSR LEFT);  
 .BY #53F, 'R', 'E', 'D', 'O', #500;  
 registro memorizzazione 'flag' scrolling;  
 registro memorizzazione NC+Y.  
 ROUTINE INPUT CONTROLLO (Dec. 49175)  
 controlla virgola e metti in 'X' il prossimo intero (NC);  
 se NC>78  
 ILLEGAL QUANTITY ERROR,  
 altrimenti \$C000=NC;  
 prendi il prossimo carattere;  
 se ≠ da 'virgolette'  
 e da ',';  
 SYNTAX ERROR;  
 se = 'virgolette', crea un indicatore del messaggio;  
 se il prossimo carattere non è ',';  
 altrimenti stampa il messaggio;  
 se in modo diretto, ILLEGAL DIRECT ERROR;  
 metti una ',' prima della stringa in ingresso;  
 definisci la variabile e metti il suo puntatore in A,Y;  
 \$49,\$4A=puntatore alla variabile;  
 'X','Y'=posizione del cursore;  
 azzerà il 'flag' dello scrolling;  
 \$C0015=NC+Y;  
 se NC+Y>78  
 \$C0015=78 (max. NC+Y ammesso);  
 NC=78-Y;  
 se cursore in riga 25,  
 e cursore in colonna 40,  
 incrementa il 'flag' dello scrolling;  
 se la prima linea sullo schermo è di 80 caratteri,  
 incrementa ancora il 'flag' dello scrolling;  
 stampa 'RVS ON';  
 stampa in 'reverse' il campo di NC caratteri;  
 cancella il carattere successivo al campo;  
 se c'è stato scrolling,  
 decrementa di uno il n° di riga;  
 se lo scrolling è stato di due righe,  
 decrementa di uno il n° di riga;  
 cursore ad inizio campo;  
 azzerà il numero di caratteri introdotti;  
 memorizza 'blank' nel 'flag' del cursore;  
 fissa la frequenza di lampeggio;  
 cursore in 'X'=\$C006, 'Y'=\$C005;  
 'flag' cursore in 'A';  
 se la variabile non è numerica,  
 ON/OFF il 'flag' "\$";  
 se la variabile è intera,  
 ON/OFF il 'flag' "%";  
 se la variabile è reale,  
 ON/OFF il 'flag' ".";  
 memorizza lo stato del 'flag' in \$C007,  
 stampa il 'flag';  
 prendi un carattere dal 'buffer' di tastiera;  
 se non è stato premuto alcun tasto,  
 genera un ritardo di 1 ms.,  
 112 volte;  
 dopo di che, ON/OFF il 'flag' e continua ad aspettare;  
 se il carattere non è 'CRSR R',  
 nè 'CRSR D',  
 nè 'CRSR U',  
 nè 'HOME',  
 nè 'CLR',  
 \$C08A LDA \$DA  
 \$C08C BML \$C091  
 \$C08E INC \$C014  
 \$C091 LDA #512  
 \$C093 JSR \$FFD2  
 \$C096 LDY \$C000  
 \$C099 LDA #520  
 \$C09B JSR \$FFD2  
 \$C09E DEY  
 \$C09F BNE \$C098  
 \$C0A1 LDY #503  
 \$C0A3 LDA \$C007,X  
 \$C0A6 JSR \$FFD2  
 \$C0A9 DEX  
 \$C0AA BNE \$C0A3  
 \$C0AC LDA \$C014  
 \$C0AF BEQ \$C0C2  
 \$C0B1 DEC \$C002  
 \$C0B4 DEC \$C006  
 \$C0B7 CMP #502  
 \$C0B9 BNE \$C0C2  
 \$C0BB DEC \$C002  
 \$C0BE DEC \$C006  
 \$C0C1 NOP  
 \$C0C2 CLC  
 \$C0C3 LDY \$C002  
 \$C0C6 LDY \$C001  
 \$C0C9 JSR \$FFFO  
 \$C0CC LDY #500  
 \$C0CE STX \$C003  
 \$C0D1 LDA #520  
 \$C0D3 STA \$C007  
 \$C0D6 LDY #570  
 \$C0D8 STY \$C004  
 \$C0DB CLC  
 \$C0DC LDY \$C006  
 \$C0DF LDY \$C005  
 \$C0E2 JSR \$FFFO  
 \$C0E5 LDA \$C007  
 \$C0E8 LDY #00  
 \$C0EA BEQ \$C0F0  
 \$C0EC EOR #504  
 \$C0EE BNE \$C0FA  
 \$C0F0 LDY \$0E  
 \$C0F2 BEQ \$C0F8  
 \$C0F4 EOR #505  
 \$C0F6 BNE \$C0FA  
 \$C0F8 EOR #50E  
 \$C0FA STA \$C007  
 \$C0FD JSR \$FFD2  
 \$C100 JSR \$E124  
 \$C103 CMP #500  
 \$C105 BNE \$C111  
 \$C107 JSR \$EEB3  
 \$C10A DEC \$C004  
 \$C10D BNE \$C100  
 \$C10F BEQ \$C006  
 \$C111 CMP #510  
 \$C113 BEQ \$C183  
 \$C115 CMP #511  
 \$C117 BEQ \$C183  
 \$C119 CMP #591  
 \$C11B BEQ \$C183  
 \$C11D CMP #513  
 \$C11F BEQ \$C183  
 \$C121 CMP #593  
 \$C123 BEQ \$C183

## COMMODORE 64

per ricominciare la procedura di immissione.

Si tenga inoltre presente che:

- durante la procedura di input sono interdetti i tasti Home,Clr,Crsr U, Crsr D, Crsr R, Del, Inst. È invece possibile servirsi del tasto Crsr L (salvo che non ci si trovi all'inizio del campo) per tornare a correggere caratteri errati. Il tentativo di servirsi dei tasti proibiti, genera un impulso sonoro, cancella il contenuto del campo in reverse e riporta il cursore all'inizio del campo per ricominciare la procedura di immissione;

- se <char.numb.> è maggiore di 78 viene segnalato un "Illegal Quantity Error";

- lo spazio totale interessato alla procedura di input, non può superare le due righe fisiche; ad esempio, se al momento del richiamo della routine Cipt il cursore si trova alla riga 5, colonna 30, il messaggio di "prompt" è composto di 6 caratteri e <char.numb.> ha valore 50, la routine assegna a <char.numb.> il nuovo valore 42 (= 78-30-6);

- se si tenta di richiamare la routine in modo diretto, viene segnalato un "Ille-

gal Direct Error".

Si veda il prossimo paragrafo per un semplice esempio di impiego di questa routine.

★ **SYS PLOT**, <column>, <row>

È una semplice e nota routine che posiziona il cursore alla colonna <column> e alla riga <row>, dove <column> può assumere valori compresi fra 0 e 39, e <row> valori compresi fra 0 e 24.

Ad esempio l'esecuzione della linea:

```
PLOT = 49852:A = 17:SYS PLOT,A,11:
PRINT "ECCO"
```

stampa Ecco al centro dello schermo.

★ **SYS BIP**

Genera un impulso sonoro; è la stessa routine già apparsa su **Personal Software** del Marzo '85 pagina 86.

Ad esempio l'esecuzione della linea:

```
BIP = 49780:SYS BIP
```

produce un "beep".

★ **SYSEL**, <column>, <row>

Cancella lo schermo dalla posizione

<row>, <column> a fine linea fisica; i limiti per <column> e <row> sono i consueti (0 ÷ 39 e 0 ÷ 24).

Ad esempio, si cancelli lo schermo e sulla prima riga si diti:

```
EL = 49859:R% = 0:C% = 20:SYSEL,
C%,R%
```

L'esecuzione di questa linea cancella dallo schermo la metà destra della linea stessa appena digitata.

★ **SYSEF**, <column>, <row>

Cancella lo schermo dalla posizione <row>, <column> a fine pagina; i limiti per <column> e <row> sono i consueti (0 ÷ 39 e 0 ÷ 24).

Ad esempio, dopo aver riempito a piacere le ultime due righe dello schermo, si riporti il cursore in alto a sinistra e si diti:

```
EF = 49881:SYSEF,0,23
```

L'esecuzione di questa linea cancella le ultime due righe dello schermo.

★ **SYSCL**, <col.1>, <row1>, <col.2>, <row2>

Cancella lo schermo dalla posizione

Seguito listato disassemblato.

```
$C125 CMP #514
$C127 BEQ $C183
$C129 CMP #594
$C12B BEQ $C183
$C12D PHA
$C12E LDX #503
$C130 LDA $C00A,X
$C133 JSR $FFD2
$C136 DEX
$C137 BNE $C130
$C139 PLA
$C13A CMP #500
$C13C BEQ $C19C
$C13E JSR $E716
$C141 CMP #590
$C143 BNE $C158
$C145 LDA #512
$C147 JSR $FFD2
$C14A LDX $C005
$C14D CPX $C001
$C150 BEQ $C183
$C152 DEC $C003
$C155 JMP $C190
$C158 LDX $C003
$C15B STA $200,X
$C15E INX
$C15F STX $C003
$C162 CPX $C000
$C165 BEQ $C16B
$C167 BCS $C183
$C169 BCC $C170
$C16B LDA #592
$C16D JSR $FFD2
$C170 SEC
$C171 JSR $FFD0
$C174 TYA
$C175 CPY #528
$C177 BNE $C17A
$C179 DEX
$C17A STX $C006
$C17D STY $C005
$C180 JMP $C001
$C183 JSR $C274
$C186 LDX $C002
$C189 LDY $C001
$C18C CLC
$C18D JMP $C04E
$C190 LDX $D6
$C192 LDY $D3
$C194 CPY #527
$C196 BNE $C199
$C198 INX
$C199 JMP $C17A
$C19C LDX $C003
$C19F LDA #500
$C1A1 STA $D200,X
$C1A4 LDX $C002
$C1A7 CLC
$C1A8 LDY $C001
$C1AB JSR $FFD0
$C1AE LDA #592
$C1B0 JSR $FFD2
$C1B3 LDX $C003
$C1B6 BEQ $C1C4
$C1B8 LDY #500
```

cancella il carattere precedente;  
 se il carattere è ≠ da 'CR',  
 genera l'eco del carattere introdotto;  
 se si tratta di 'CRSR L',  
 stampa un 'RVS ON';  
 se il cursore non è all'inizio del campo,  
 decrementa il n° di caratteri inseriti,  
 e salta a \$C190;  
 se il carattere non è 'CRSR L',  
 memorizzalo nell'input buffer';  
 incrementa il n° di caratteri inseriti;  
 se era l'ultimo carattere del campo, stampa 'RVS ON';  
 se non si è superato il n° di caratteri del campo (NC),  
 controlla la posizione del cursore;  
 se ha cambiato riga,  
 decrementa il n° di riga,  
 \$C006, \$C005=posizione corrente del cursore;  
 preparati a ricevere il prossimo carattere;  
 se si tratta di un tasto "proibito", oppure  
 si è superata la lunghezza del campo (NC),  
 emetti un 'beep'  
 e ricomincia;  
 se si tratta di 'CRSR L' e  
 ci si trova alla fine della riga,  
 incrementa il n° di riga;  
 salta a \$C17A;  
 se il carattere è 'CR',  
 forza uno #500 nell'input buffer';  
 posiziona il cursore all'inizio del campo;  
 stampa 'RVS OFF';  
 se è stato introdotto almeno un carattere,

## Seguito listato disassemblato.

\$C1BA LDA \$0200, Y  
 \$C1B0 JSR \$FFD2  
 \$C100 INY  
 \$C1C1 DEX  
 \$C1C2 BNE \$C1BA  
 \$C1C4 SEC  
 \$C1C5 LDA \$0000  
 \$C1C8 SBC \$0003  
 \$C1C8 TAX  
 \$C1C8 BEQ \$C1D6  
 \$C1CE LDA #\$20  
 \$C1D0 JSR \$FFD2  
 \$C1D3 DEX  
 \$C1D4 BNE \$C1D0  
 \$C1D6 LDA \$7A  
 \$C1D8 LDY \$7B  
 \$C1DA STA \$4B  
 \$C1DC STY \$4C  
 \$C1DE LDY #\$FF  
 \$C1E0 LDY #\$01  
 \$C1E2 STX \$7A  
 \$C1E4 STY \$7B  
 \$C1E6 JSR \$0073  
 \$C1E9 BIT \$00  
 \$C1EB BPL \$C1FD  
 \$C1ED LDA \$7A  
 \$C1EF LDY \$7B  
 \$C1F1 JSR \$848D  
 \$C1FA JSR \$87E2  
 \$C1F7 JSR \$A9DA  
 \$C1FA JMP \$C205  
 \$C1FD JSR \$86F3  
 \$C200 LDA \$0E  
 \$C202 JSR \$A9C2  
 \$C205 JSR \$0079  
 \$C208 BEQ \$C26B  
 \$C20A LDA \$00  
 \$C20C PHA  
 \$C20D LDA \$0E  
 \$C20F PHA  
 \$C210 CLC  
 \$C211 LDY \$0002  
 \$C214 LDY \$0001  
 \$C217 JSR \$FFFO  
 \$C21A JSR \$C27A  
 \$C21D LDY \$0000  
 \$C220 LDA #\$20  
 \$C222 JSR \$FFD2  
 \$C225 DEX  
 \$C226 BNE \$C222  
 \$C228 CLC  
 \$C229 LDY \$0002  
 \$C22C LDY \$0001  
 \$C22F JSR \$FFFO  
 \$C232 LDA \$0E  
 \$C234 LDY \$00  
 \$C236 JSR \$AB1E  
 \$C239 LDY #\$04  
 \$C23B LDY #\$00  
 \$C23D JSR \$EEB3  
 \$C240 DEX  
 \$C241 BNE \$C23D  
 \$C243 DEY  
 \$C244 BNE \$C23B

esegui l'eco in 'normal' dei caratteri introdotti,  
 e cancella il resto del campo;  
 \$4B, \$4C=puntatore al carattere corrente;  
 \$7A, \$7B=puntatore all'input buffer';  
 prendi il prossimo carattere dall'input buffer';  
 se si tratta di una variabile non numerica,  
 carica l'indicatore della stringa nell'accum. f1p;  
 metti il puntatore al testo nel ptr al carattere corrente;  
 metti l'indicatore della stringa nella variabile stringa;  
 se la variabile è numerica, converti la stringa ASCII  
 in f1p;  
 carica il 'flag' "intero";  
 metti l'accumulatore nella variabile numerica;  
 prendi il carattere corrente;  
 se ≠ 0,  
 salva i 'flags' di "tipo variabile" nello stack;  
 cursore all'inizio del campo;  
 emetti un 'beep';  
 cancella il dato errato;  
 cursore all'inizio del campo;  
 stampa '? REDO'  
 per circa 1 sec.;

\$C246 CLC  
 \$C247 LDY \$0002  
 \$C24A LDY \$0001  
 \$C24D JSR \$FFFO  
 \$C250 LDY \$05  
 \$C252 LDA #\$20  
 \$C254 JSR \$FFD2  
 \$C257 DEX  
 \$C258 BNE \$C246  
 \$C25A PLA  
 \$C25B STA \$0E  
 \$C25D PLA  
 \$C25E STA \$00  
 \$C260 LDA \$4B  
 \$C262 LDY \$4C  
 \$C264 STA \$7A  
 \$C266 STY \$7B  
 \$C268 JMP \$C186  
 \$C26B LDA \$4B  
 \$C26D LDY \$4C  
 \$C26F STA \$7A  
 \$C271 STY \$7B  
 \$C273 RTS

cursore all'inizio del campo;  
 cancella il messaggio;  
 ripristina i 'flags' di "tipo variabile";  
 ripristina il puntatore al carattere corrente;  
 salta a \$C186;  
 ripristina il puntatore al carattere corrente  
 e ritorna.

ROUTINE BEEP (Dec. 49780)  
 azzeramento  
 del SID;  
 fissa la pulse  
 waveform width;  
 fissa  
 Attack/Decay;  
 fissa la frequenza  
 (byte alto);  
 seleziona la pulse  
 waveform e setta Att/Dec/Sus;  
 volume al  
 massimo.

ROUTINE INPUT 2 PARAMETRI  
 controlla virgola e metti in 'X' il prossimo intero  
 (colonna);  
 se > 29,  
 ILLEGAL QUANTITY ERROR;  
 salva l'intero nello stack;  
 controlla virgola e metti in 'X' il prossimo intero  
 (riga);  
 se > 24, ILLEGAL QUANTITY ERROR;  
 metti in 'Y' il primo intero (colonna)  
 e ritorna.

ROUTINE PLOT (Dec. 49852)  
 leggi i due parametri, mettili in 'X', 'Y'  
 e posiziona il cursore in 'X' (riga), 'Y' (colonna).

\$C274 LDA #\$00  
 \$C276 STA \$D404  
 \$C279 STA \$D405  
 \$C27C STA \$0406  
 \$C27F STA \$D402  
 \$C282 LDA \$07  
 \$C284 STA \$D403  
 \$C287 LDA \$22  
 \$C289 STA \$0405  
 \$C28C LDA \$52  
 \$C28E STA \$D401  
 \$C291 LDA \$41  
 \$C293 STA \$D404  
 \$C296 LDA \$0F  
 \$C298 STA \$D418  
 \$C29B RTS  
 \$C29C NOP  
 \$C29D NOP  
 \$C29E LDA #\$28  
 \$C2A0 STA \$FE  
 \$C2A2 LDA #\$19  
 \$C2A4 STA \$FD  
 \$C2A6 JSR \$B7F1  
 \$C2A9 CPX #\$28  
 \$C2AB BCC \$C2B0  
 \$C2AD JMP \$B248  
 \$C2B0 TXA  
 \$C2B1 PHA  
 \$C2B2 JSR \$B7F1  
 \$C2B5 CPX #\$19  
 \$C2B7 BCS \$C2AD  
 \$C2B9 PLA  
 \$C2BA TAY  
 \$C2BB RTS

\$C2BC JSR \$C29E  
 \$C2BF JSR \$FFFO  
 \$C2C2 RTS

## COMMODORE 64

## Seguito listato disassemblato.

```

ROUTINE EL (End of Line - Dec. 49859)
leggi due parametri e mettili in 'X', 'Y';
leggi l'indirizzo dello schermo;
leggi l'indirizzo della memoria colore;
cancella un byte sul video;
cancella un byte nella memoria colore;
ripeti fino alla colonna $FE;
ritorna.

ROUTINE EF (End of Frame - Dec. 49881)
leggi due parametri e mettili in 'X', 'Y';
esegui la routine EL;
cancella tutte le righe fino alla
24 compresa
e ritorna.

ROUTINE CL (Clear - Dec. 49895)
leggi due parametri e mettili in 'X', 'Y'
(CI, RI);

salvali nello stack;
leggi due parametri e mettili in 'X', 'Y'
(C2, R2);
salvali in $FB, $FC;
metti C2-1 in $FD;
ripulisti CI, RI in 'X', 'Y';
se RI>R2, ILLEGAL QUANTITY ERROR;

se RI=R2 e CI>C2, ILLEGAL QUANTITY ERROR;
cancella le righe da RI+1 a R2-1;
se RI=R2 cancella da inizio riga a C2,
altrimenti
cancella da CI a C2,
e ritorna.

```

## Listato 2 - Il programma Easy Video 64.

```

0 REM *****
1 REM *
2 REM *** EASY VIDEO PER C.64 ***
3 REM * ----- *
4 REM * DI *
5 REM * MAURIZIO PAOLINELLI *
6 REM * VIA MAGELLANO, 15 *
7 REM * 20094 CORSICO MI *
8 REM * TEL. (02)4407707 *
9 REM *
10 REM*****
16 GOSUB80
18 PRINT"{GIU'}ATTENDI CIRCA 10 SECONDI,
PREGO!"
19 REM *** LETTURA DELLE NUOVE ROUTINE
20 A=0:FORI=49152TO49946:READN:A=A+N:POKE
I,N:NEXT
22 IFA<101666THENGOSUB80:PRINT"
{ 4 GIU'}ERRORE NELLE ISTRUZIONI DATA!
":END
24 GOSUB80:PRINTSPC(7)CHR$(14)"{GIU'}SINT
ASSI DEI NUOVI COMANDI"
26 PRINTSPC(7)"-----
{GIU'}"
28 PRINT"CIPT=49175"TAB(20)"EL=49859":PRI
NT"PLOT=49852"TAB(20)"EF=49881"
30 PRINT"BIP=49780"TAB(20)"CL=49895"
32 PRINT"{ 2 GIU'}SYSCRIPT,<CHAR.NUMB.>[<C
HR$(34)"<PROMPT>"CHR$(34)"];<VAR.>"
34 PRINT"SYSLOT,<COLUMN>,<ROW>"
36 PRINT"SYSBIP"
38 PRINT"SYSEL,<COLUMN>,<ROW>"
40 PRINT"SYSEF,<COLUMN>,<ROW>"
42 PRINT"SYSCL,<COL.1>,<ROW1>,<COL.2>,<RO
W2>"
44 NEW:END
80 PRINT"{CLR}{GIU'}"SPC(6):FORI=0TO26:PR
INT"*";:NEXT:PRINT
82 PRINTSPC(6)"*** EASY VIDEO PER C.64 **
*"
84 PRINTSPC(6):FORI=0TO26:PRINT"*";:NEXT:
PRINT:RETURN
100 DATA3,6,1,0,69,6,1,32,18,32,146,157,3
2,157,63,82,69
110 DATA68,79,0,0,15,234,32,241,183,224,7
9,144,3,76,72,178,142
120 DATA0,192,32,121,0,201,34,240,8,169,5
9,32,255,174,76,62,192
130 DATA32,189,174,169,59,32,255,174,32,3
3,171,32,166,179,169,44,141
140 DATA255,1,32,139,176,133,73,132,74,56
,32,240,255,142,2,192,142
150 DATA6,192,140,1,192,140,5,192,169,0,1
41,20,192,24,173,0,192
160 DATA109,1,192,141,21,192,201,79,144,1
2,56,169,78,141,21,192,237
170 DATA1,192,141,0,192,224,24,144,17,173
,21,192,201,39,144,10,238
180 DATA20,192,165,218,48,3,238,20,192,16
9,18,32,210,255,172,0,192
190 DATA169,32,32,210,255,136,208,250,162

```

## COMMODORE 64

<row1>, <col.1> alla posizione <row2>, <col.2>; i limiti per <col.1>, <col.2>, <row1>, <row2> sono i consueti (0 ÷ 39 e 0 ÷ 24). Se il punto iniziale <row1>, <col.1>, si trova successivamente al punto finale <row2>, <col.2>, viene segnalato un "Illegal Quantity Error".

Ad esempio, si riempiono le prime tre righe dello schermo con caratteri a piacere, si preme almeno una volta Crsr D; e si digiti:

```
CL = 49895:R1 = 1:C1 = 8:SYSCL,C1,R1,
C1-1,R1+1
```

L'esecuzione di questa linea cancella dallo schermo 40 caratteri a partire dalla seconda riga, 9° colonna.

L'esecuzione della linea:

```
CL = 49895:SYSCL,3,2,1
```

provoca un "Illegal Quantity Error" poiché il punto iniziale (2,3) si trova dopo il punto finale (1,3).

### Un esempio di applicazione

Come semplice esempio di utilizzo delle routine descritte, si digiti il seguente programma:

```
NEW <RETURN>
10 CIPT = 49175:PLOT = 49852:
   BIP = 49780:EL = 49859:PRINT
   "[<CLR>]"
15 SYSLOT,10,2:PRINT"QUESTO
   È UN ESEMPIO"
20 SYSLOT,0,10:SYSCRIPT,5
   "DATO (> 13000): ";A
30 IFA <= 13000 THEN
   SYSBIP:SYSEL,0,10:GOTO 20
40 SYSLOT,0,20:PRINT"HAI
   INTRODOTTTO IL NUMERO";A:END
RUN <RETURN>
```

Dopo aver stampato sulla terza riga dello schermo "Questo è un esempio", alla riga 10 compare il messaggio Dato (> 13000): e un campo in reverse di 5 caratteri con un punto lampeggiante sul 1° carattere.

Si premiano i tasti 123456 e si osservi l'eco sul video. Quando si preme il tasto 6 (si cerca di immettere il 6° carattere in un campo definito di 5 caratteri), si sente un "beep", il campo viene cancellato e si riposiziona il cursore sul 1° carattere per una nuova introduzione.

Si premiano i tasti 123A <Return>. Si cerchi cioè di introdurre nella variabile numerica A un dato alfanumerico. Si sente un "beep", compare il messaggio "? Redo" per un secondo e quindi si ricomincia dall'inizio.

Si premiano ora i tasti 12345 <Return>. Essendo il numero introdotto minore di 13.000, viene cancellata l'intera riga n° 10 e ripresentata la richiesta di immissione.

Si premiano i tasti 129 <Crsr L> <Crsr L> 3201 <Return>. Sul video compare il numero corretto 13.201 al posto del campo in reverse e alla riga 20 il messaggio definito nella linea 40 del programma. ■

#### Seguito programma Easy Video 64.

```

,3,189,7,192,32,210,255,202
200 DATA208,247,173,20,192,240,17,206,2,1
   92,206,6,192,201,2,208,7
210 DATA206,2,192,206,6,192,234,24,174,2,
   192,172,1,192,32,240,255
220 DATA162,0,142,3,192,169,32,141,7,192,
   160,112,140,4,192,24,174
230 DATA6,192,172,5,192,32,240,255,173,7,
   192,166,13,240,4,73,4
240 DATA208,10,166,14,240,4,73,5,208,2,73
   ,14,141,7,192,32,210
250 DATA255,32,36,225,201,0,208,10,32,179
   ,238,206,4,192,208,241,240
260 DATA197,201,29,240,110,201,17,240,106
   ,201,145,240,102,201,19,240,98
270 DATA201,147,240,94,201,20,240,90,201,
   148,240,86,72,162,3,189,10
280 DATA192,32,210,255,202,208,247,104,20
   1,13,240,94,32,22,231,201,157
290 DATA208,19,169,18,32,210,255,174,5,19
   2,236,1,192,240,49,206,3
300 DATA192,76,144,193,174,3,192,157,0,2,
   232,142,3,192,236,0,192
310 DATA240,4,176,26,144,5,169,146,32,210
   ,255,56,32,240,255,152,192
320 DATA40,208,1,202,142,6,192,140,5,192,
   76,209,192,32,116,194,174
330 DATA2,192,172,1,192,24,76,78,192,166,
   214,164,211,192,39,208,1
340 DATA232,76,122,193,174,3,192,169,0,15
   7,0,2,174,2,192,24,172
350 DATA1,192,32,240,255,169,146,32,210,2
   55,174,3,192,240,12,160,0
360 DATA185,0,2,32,210,255,200,202,208,24
   6,56,173,0,192,237,3,192
370 DATA170,240,8,169,32,32,210,255,202,2
   08,250,165,122,164,123,133,75
380 DATA132,76,162,255,160,1,134,122,132,
   123,32,115,0,36,13,16,16
390 DATA165,122,164,123,32,141,180,32,226
   ,183,32,218,169,76,5,194,32
400 DATA243,188,165,14,32,194,169,32,121,
   0,240,97,165,13,72,165,14
410 DATA72,24,174,2,192,172,1,192,32,240,
   255,32,116,194,174,0,192
420 DATA169,32,32,210,255,202,208,250,24,
   174,2,192,172,1,192,32,240
430 DATA255,169,14,160,192,32,30,171,160,
   4,162,0,32,179,238,202,208
440 DATA250,136,208,245,24,174,2,192,172,
   1,192,32,240,255,162,5,169
450 DATA32,32,210,255,202,208,250,104,133
   ,14,104,133,13,165,75,164,76
460 DATA133,122,132,123,76,134,193,165,75
   ,164,76,133,122,132,123,96,169
470 DATA0,141,4,212,141,5,212,141,6,212,1
   41,2,212,169,7,141,3
480 DATA212,169,34,141,5,212,169,82,141,1
   ,212,169,65,141,4,212,169
490 DATA15,141,24,212,96,234,234,169,40,1
   33,254,169,25,133,253,32,241
500 DATA183,224,40,144,3,76,72,178,138,72
   ,32,241,183,224,25,176,244
510 DATA104,168,96,32,158,194,32,240,255,
   96,32,158,194,32,240,233,32
520 DATA36,234,169,32,145,209,32,218,228,
   200,196,254,144,244,96,32,158
530 DATA194,32,198,194,160,0,232,228,253,
   144,246,96,32,158,194,138,72
540 DATA152,72,32,166,194,134,251,202,134
   ,253,132,252,104,168,104,170,228
550 DATA251,240,4,176,171,208,8,196,252,2
   40,9,176,163,144,5,32,220
560 DATA194,160,0,165,252,133,254,230,254
   ,32,198,194,96
```

# Cubo Magico

Giochiamo con lo Spectrum ad una nuova versione del celeberrimo rompicapo

di Compagiani

L'idea del programma è nata dalla curiosità di vedere se riuscivamo a realizzare un gioco "manuale" su di un piccolo elaboratore.

Naturalmente il programma ha delle limitazioni rispetto al gioco originale (non pensate di poter battere il record mondiale di velocità nel risolverlo), ma grazie alle indiscutibili doti del nostro Spectrum, abbiamo raggiunto delle varianti al gioco, varianti che sarebbero risultate impossibili da realizzare con il vero cubo. Così il cubo non è più 3+3, ma 4+4. I puristi del cubo di Rubik non gridino allo scandalo: il motivo di questo cambiamento rivoluzionario (che scompagina tutte le tattiche esistenti) sta nel fatto che vogliamo lasciare la possibilità anche ai neofiti di questo gioco, di elaborare una propria strategia, o per lo meno di non sfigurare di fronte agli esperti. Se un "purista" si è sentito male dopo questa mini-rivoluzione, non legga avanti.

Infatti le variazioni sull'originale non sono finite: quando facciamo scorrere le file più esterne, cosa che normalmente farebbe girare su se stessa anche la faccia antistante, questa ultima in realtà non si gira su se stessa.

In definitiva potremo soprannominare il programma "Cubo digitale", tante sono le variazioni (figura 1).

Il programma ha avuto una lunghissima fase di debug per cui non demoralizzatevi se qualche routine vi risulta un po' oscura: in definitiva il programma è risultato piacevole, con una buona grafica, abbastanza veloce, effetti musicali, e senza alcun bug possibile.

Finora siamo riusciti a risolvere il cubo al livello 1. Per curiosità vi informiamo che dopo la scelta del livello, il mischiamento del cubo è dovuto sia al numero che avete scelto, che a un coefficiente casuale. Perciò ci può essere un livello 3 più difficile di un livello 4.

In conclusione il programma vuole essere uno stimolo per la vostra intelligenza, troppo lungo intorpidita dai giochi "spara e fuggi" dello Spectrum;

## Caricamento e funzionamento

Il programma è diviso in due parti:

- 1) caricatore BASIC;
- 2) programma vero e proprio.

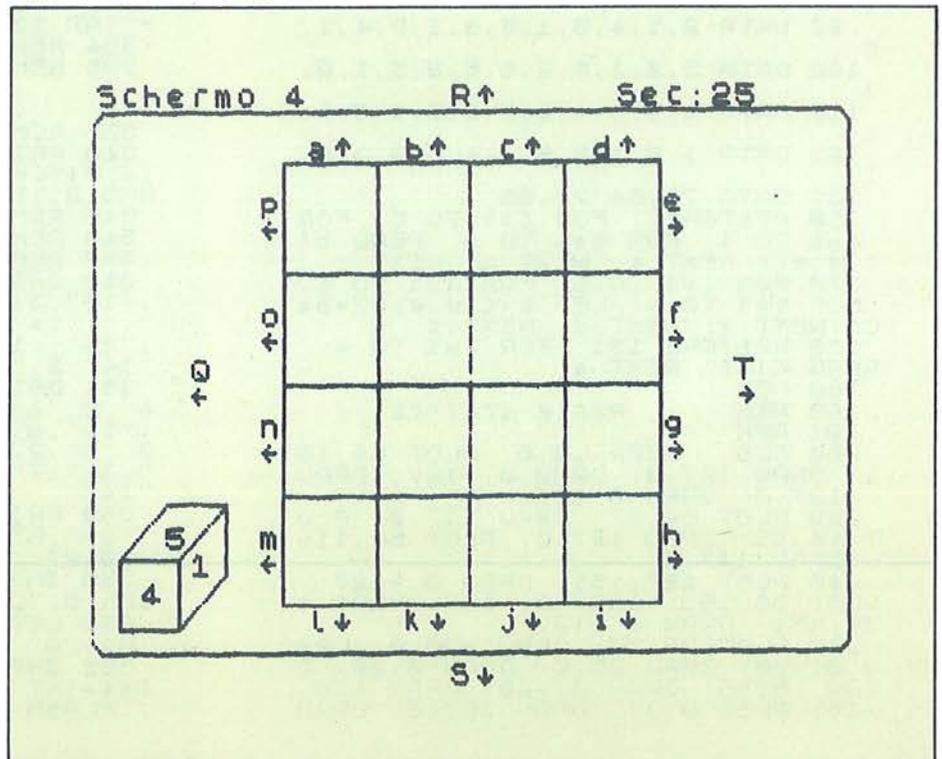
Il caricatore BASIC definisce i 4 caratteri grafici del programma, le frecce, carica in un'area di memoria riservata (da 65000) il linguaggio macchina necessario (44 byte) e con Load"" carica il vero programma. Da notare la linea "0" ottenuta "pokando" in 23755,0. La linea "0" è anche usata da software house professionali per la protezione dei programmi (Edit impossibile, Merge che non funziona). In fase di battitura la si può anche omettere. Per salvare il cari-

catore battere in modo immediato "Go-to 9999". Infatti a questa linea si salva il programma con l'AutoRUN(Line).

Dopo aver salvato il caricatore lo si può anche cancellare e passare a battere il programma vero e proprio. Per battere quest'ultimo non ci dovrebbero essere problemi. Forse l'unica difficoltà potrebbe derivare dai caratteri grafici. Vi ricordiamo che la A è la freccia in alto, B in basso, C verso sinistra e D verso destra.

Dopo aver battuto tutto, lo si deve regi-

Figura 1 - Una videata del gioco.



strare subito dopo il caricatore, lasciando solo un piccolo spazio.

Per far ciò battete in modo diretto Save "Line 1.

Predisponete i cavetti e eseguite quello che avrete già fatto decine di volte. Evitate di fare "pastrocchi" sovrapponendo il programma al caricatore. Se tutto OK possiamo provare il programma e vedere come funziona. Riavvolgete il nastro a prima del caricatore e date il Load (oppure date Run se avete la routine in linguaggio macchina già in memoria). Dopo un attimo di attesa vi verrà richiesto il livello di gioco. Vi consigliamo di battere 0, almeno per le prime

volte.

Il computer comincerà a mischiare il cubo mostrandovi la situazione sul video. Quando avrà finito premete Enter per iniziare. Premendo un tasto da "a" a "p" potrete spostare la fila scelta nella direzione voluta. Ad esempio premendo "e" cambierà tutta la prima fila. Premendo "Q", "R", "S", "T" cambierà il punto di osservazione del cubo. Da notare in basso a sinistra la piccola rappresentazione tridimensionale del cubo.

Nella parte alta dello schermo, a sinistra è indicato il numero di schermo mostrato (per cambiarlo, basta premere Q,R,S,T), mentre a destra scorre il tem-

po. Quando ritenete di aver completato il cubo, battete Enter; se lo avete completato veramente comparirà un messaggio di complimentazione accompagnato da una musica, in seguito verrà scritto il tempo impiegato per completarlo. Vi viene chiesto se volete rigiocare (S/N) e ad una risposta negativa il computer si resetta (New).

Se invece non avevate completato il cubo, il computer ve lo farà notare, e successivamente vi verrà chiesto se volete continuare. Rispondendo "S" potrete continuare a sistemare il cubo, mentre rispondendo "N" potrete ricominciare una nuova partita.

#### Listato 1 - Il programma Cubo Magico.

```

15 BEEP .2,0
20 LET SCHERMO=1: BORDER 0: PA
PER 0: INK 6: CLS : PRINT #0; PA
PER 4; INK 0; FLASH 1; " Fer
ma il registratore
70 DIM a(6,4,4): DIM B(6,4,3):
DIM G(4): DIM H(4)
71 REM
Dati per le 6 faccie
72 DATA 4,3,2,5,3,6,6,3,5,2,3,
4
82 DATA 1,4,3,5,4,6,6,4,5,3,4,
1
92 DATA 2,1,4,5,1,6,6,1,5,4,1,
2
102 DATA 3,2,1,5,2,6,6,2,5,1,2,
3
112 DATA 1,6,3,4,6,2,2,6,4,3,6,
1
122 DATA 1,5,3,2,5,4,4,5,2,3,5,
1
131 DATA 76,64,72,68
160 RESTORE : FOR c=1 TO 6: FOR
d=1 TO 4: FOR e=1 TO 3: READ b(
c,d,e): NEXT e: NEXT d: NEXT c
170 FOR c=1 TO 6: FOR d=1 TO 4:
FOR e=1 TO 3: LET a(c,d,e)=7+8*
C: NEXT e: NEXT d: NEXT c
180 RESTORE 131: FOR a=1 TO 4:
READ h(a): NEXT a
189 REM
190 REM Parte grafica
191 REM
200 CLS : BEEP .2,0: PLOT 64,15
1: DRAW 127,0: DRAW 0,-127: DRAW
-127,0: DRAW 0,127
220 PLOT 64,87: DRAW 127,0: PLO
T 64,55: DRAW 127,0: PLOT 64,119
: DRAW 127,0
240 PLOT 127,151: DRAW 0,-127:
PLOT 96,151: DRAW 0,-127: PLOT 1
59,151: DRAW 0,-127
260 PLOT 29,37: DRAW -20,0: DRA
W 0,-20: DRAW 20,0: DRAW 0,20: D
RAW 16,16: DRAW 0,-20: DRAW -16,
-16: PLOT 9,37: DRAW 16,16: DRAW

```

```

20,0
270 PLOT 1,12: PLOT 2,11: PLOT
3,10: DRAW 249,0: DRAW 3,3: DRAW
0,150: DRAW -3,3: DRAW -249,0:
DRAW -3,-3: DRAW 0,-150
275 REM
276 REM Tasti per muovere
277 REM
280 PRINT AT 2,9;"a↑ b↑ c↑ d
↑";AT 19,9;"l↓ k↓ j↓ i↓"
300 PRINT AT 4,7;"p";AT 8,7;"o"
;AT 12,7;"n";AT 16,7;"m"
310 PRINT AT 5,7;"+";AT 9,7;"+"
;AT 13,7;"+";AT 17,7;"+"
320 PRINT AT 4,24;"e";AT 8,24;"
f";AT 12,24;"g";AT 16,24;"h"
330 PRINT AT 5,24;">";AT 9,24;"
>";AT 13,24;">";AT 17,24;">"
334 REM
335 REM Scrive il numero della
faccia e i numeri delle
facce laterali ad essa
336 REM
340 PRINT AT 18,2;SCHERMO;AT 17
,4;B(SCHERMO,4,1);AT 16,3;B(SCHE
RMO,2,1)
345 REM
346 REM Dati per la musica
347 REM
350 DATA 2,"15",1,"15",3,"16",3
,"15",3,"17",3,"14",2,"13",5,"12
",1,"14",1,"17",2,"19",4,"21",1
,"36",1,"36",1,"36",1,"36",1,
"14",2,"17",2,"19",4,"21",1,
351 DATA 0,"0",0,"0",0,"0",1,"1
4",2,"17",2,"19",4,"21",3,"17",3
0,"0",2,"17",2,"17",3,"16",5,"14
",3,"17",2,"21",4,"19",5,"16",7,
"14"
352 PRINT AT 0,15;"R↑";AT 21,15
;"S↓";AT 10,4;"0";AT 11,4;"<";AT
10,27;"T";AT 11,27;">"
353 INPUT "": PRINT #0;AT 1,9;
INK 6;"LIVELLO?(0 - 9)"
354 LET N%=INKEY$: IF N%<"0" OR
N%>"9" THEN GO TO 354
355 INPUT "": LET J=((VAL N%)+1
)*4+INT (RND*4): PRINT #0;AT 1,0
; FLASH 1; INK 4; PAPER 0;" As

```

## ZX SPECTRUM

Seguito programma Cubo Magico

```

petta : mischio il cubo " : FO
R a=-12 TO 36: BEEP .1,a: BEEP .
01,36: NEXT a: PRINT #0;AT 1,0;
FLASH 1; INK 4; PAPER 0;" !Puo
i vedere che non baro! "
355 FOR K=1 TO J: RANDOMIZE : L
ET A$=CHR$(65+INT (RND*16)): LE
T VERSO=1*(A$>"L")+2*(A$<"E")+3*
(A$>"H" AND A$<"M")+4*(A$>"D" AN
D A$<"I")
357 LET FACCIA=SCHERMO: LET NUM
ERO=CODE A$-H(VERSO): IF (VERSO=
1 OR VERSO=3) THEN LET NUMERO=5-
NUMERO
358 GO SUB 1000+(VERSO=2 OR VER
SO=3)*100: BEEP .01,36: BEEP .01
,24: NEXT K
360 PRINT #0;AT 1,0; FLASH 1; I
NK 4; PAPER 0;" Premi ENTER
per iniziare "
365 REM
366 REM Suona la musica per una
durata=Max
367 REM
370 LET max=8
380 RESTORE 350: FOR a=1 TO max
390 IF INKEY$=CHR$ 13 THEN GO T
O 450
400 READ g,H$: BEEP g/10,VAL H$
: NEXT a
410 IF max=8 THEN LET ma=43
420 IF max=43 THEN LET ma=8
430 LET max=ma: GO TO 380
440 REM
441 REM FN b() per il tempo
442 REM
450 POKE 23658,8: POKE 23672,0:
POKE 23673,0: POKE 23674,0: DEF
FN b()=INT ((PEEK 23674*65536+P
EEK 23673*256+PEEK 23672)/50)
450 FOR A=1 TO 12: BEEP .1,A*2:
BEEP .01,36: BEEP .01,36-A*2: N
EXT A
480 BEEP .1,0: PRINT AT 18,2;SC
HERMO;AT 17,4;B(SCHERMO,4,1);AT
16,3;B(SCHERMO,2,1)
490 PRINT AT 0,0;"Schermo ";SCH
ERMO;AT 0,22;"Sec.": PRINT #0; I
NK 6;AT 1,0;"Premi ENTER per fer
mare il tempo"
495 REM
496 REM
497 REM Prende i dati dalla
INKEY$ e li passa alla
routine di gestione cubo
498 REM
499 REM
500 BEEP .01,RND*24: PRINT AT 0
,26;FN B()
520 LET A$=INKEY$: IF (A$<"A" O
R A$>"T") AND A$<>CHR$ 13 THEN G
O TO 500
530 IF A$=CHR$ 13 THEN GO TO 30
00: REM ENTER=CHR$ 13
540 IF A$>"P" THEN GO TO 2000
550 LET VERSO=1*(a$>"L" AND a$<
"0")+2*(a$>"E" AND a$<"E")+3*(a$
>"H" AND a$<"M")+4*(A$>"D" AND A
$<"I")
560 LET FACCIA=SCHERMO: LET NUM

```

```

ERO=CODE A$-H(VERSO): IF VERSO=1
OR VERSO=3 THEN LET NUMERO=5-NU
MERO
570 GO SUB 1000+(VERSO=2 OR VER
SO=3)*100: GO TO 500
990 REM
991 REM rout. di gestione cubo
in caso di spostamenti
a destra(VERSO=4) o a
sinistra(VERSO=1)
992 REM
1010 FOR N=1 TO 4: LET G(N)=A(SC
HERMO,NUMERO,N): NEXT N
1020 FOR A=1 TO 3
1030 FOR B=1 TO 4: LET FACCIA1=B
(schermo,VERSO,a)
1040 LET A(FACCIA,NUMERO,B)=A(FA
CCIA1,NUMERO,B): NEXT B
1050 LET faccia=faccia1: NEXT A
1060 FOR A=1 TO 4: LET A(FACCIA,
NUMERO,A)=G(A): NEXT A
1070 GO TO 5000
1091 REM
1092 REM rout. di gestione cubo
in caso di spostamenti
verso su'(VERSO=2) o
in giu'(VERSO=3)
1093 REM

```



## Dalla grande edicola Jackson

### Tutto sul personal computer

#### COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.  
9 numeri all'anno: L. 2.000 a numero  
Abbonamento: solo L. 15.000

#### Personal

L'unica rivista indipendente per gli utenti del personal computer Olivetti.  
10 numeri all'anno: L. 4.000 a numero  
Abbonamento: solo L. 35.000



L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.  
10 numeri all'anno: L. 5.000 a numero  
Abbonamento: solo L. 40.000

#### PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.  
11 numeri all'anno: L. 4.000 a numero  
Abbonamento: solo L. 34.000



La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato.  
11 numeri all'anno: L. 5.000 a numero  
Abbonamento: solo L. 43.000

#### Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:  
Gruppo Editoriale Jackson  
via Rosellini, 12 - 20124 MI

Desidero ricevere GRATIS un numero

della Rivista \_\_\_\_\_

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

via \_\_\_\_\_

CAP \_\_\_\_\_ Città \_\_\_\_\_

Comunque il funzionamento è molto più semplice di quanto lo si sia descritto, grazie alle frecce, che indicano senza ombra di dubbio, cosa fa un certo tasto se premuto.

Ci auguriamo che il programma risulti per lo meno interessante e di stimolo per una sempre più proficua utilizzazione dello Spectrum.

### Strutture dei dati

Per effettuare le opportune rotazioni si è dimensionato un array con misure 6,4,4, corrispondenti al numero di facce del cubo, e alle due dimensioni di ogni

faccia (base e altezza, per intenderci). Questo è l'array A (di nome e di fatto). Abbiamo poi dimensionato un array B con dimensioni 6,4,3, corrispondenti al numero di facce, alle possibili rotazioni (sinistra, sù, giù, destra), e al numero di facce coinvolte nella rotazione oltre a quella visualizzata sul video. In sostanza, per effettuare delle rotazioni, bisogna:

- 1) salvare in un array di servizio i dati che devono essere cambiati per evitare che vadano persi;
- 2) prendere dall'array B il numero dello schermo da cui prelevare i dati necessari;

3) prelevare i dati e immetterli nello schermo attuale;

4) cambiare il numero dello schermo con quello ottenuto al punto 2;

5) ritornare al punto 2 (ripetere 3 volte);

6) immettere nello schermo attuale i dati salvati nell'array di servizio al punto 1;

7) Return.

All'inizio di ogni nuova partita l'array A viene rinfrescato, cioè viene rimesso ai valori iniziali (che variano a 8 per la prima facciata, e a 48 per la sesta).

Verso la linea di programma 350 ci sono una serie di dati per la musica, che accompagna il gioco nei momenti di attesa ("Premi Enter per iniziare"), e

#### Seguito programma Cubo Magico

```

1110 FOR N=1 TO 4: LET G(N)=A(SC
HERMO,N,NUMERO): NEXT N
1120 FOR A=1 TO 3
1130 FOR B=1 TO 4: LET FACCIA1=B
(schermo,VERSO,a)
1140 LET A(FACCIA,B,NUMERO)=A(FA
CCIA1,B,NUMERO): NEXT B
1150 LET FACCIA=FACCIA1: NEXT A
1160 FOR A=1 TO 4: LET A(FACCIA,
A,NUMERO)=G(A): NEXT A
1170 GO TO 6000
1991 REM
1992 REM Cambio di vista del cu
bo
1993 REM
3000 LET VERSO=CODE A$-80
3010 LET SCHERMO=B(SCHERMO,VERSO
,1): GO SUB 6000: GO TO 480
3990 REM
3991 REM controlla se il cubo
e' finito
3992 REM
3000 LET T=PEEK 23674: LET Y=PEE
K 23673: LET U=PEEK 23672
3005 FOR A=1 TO 6: LET H=A(A,1,1
): FOR B=1 TO 4: FOR C=1 TO 4:
3010 IF A(A,B,C)<>H THEN GO TO 4
000
3020 NEXT C: NEXT B: NEXT A
3025 REM
3026 REM cubo finito
3027 REM
3030 PRINT #0;AT 1,0; INK 6;"Com
plimenti lo hai completato !!":
RESTORE 350: FOR a=1 TO 43: READ
g,h$: BEEP g/10,VAL h$: NEXT a
3040 INPUT "": POKE 23674,T: POK
E 23673,Y: POKE 23672,U: PRINT #
0;AT 1,0; INK 6;" Hai impiegato
";FN b();" secondi": FOR B=-1 T
O 3: FOR A=1 TO 20: BEEP .02,A*B
*15: NEXT A: NEXT B
3050 INPUT "": PRINT #0;AT 1,4;
INK 6;"Vuoi rigiocare? (s - n)"
3060 LET a$=INKEY$: IF a$<>"S" A
ND A$<>"N" THEN GO TO 3060
3070 IF A$="S" THEN CLS : PRINT

```

```

#0;AT 1,4; INK 6;" O.K. risistem
o il cubo ": GO TO 170
3080 NEW
3100 RESTORE 350: FOR A=1 TO 51:
READ G,H: BEEP G/10,h: NEXT A
3991 REM
3992 REM Cubo non finito
3993 REM
4000 PRINT #0;AT 1,0; INK 6;"Mi
dispiace : non completato !! ":
FOR a=1 TO 5: BEEP 1.5,30-a*6: N
EXT a
4010 INPUT "": PRINT #0;AT 1,0;
INK 6;" Vuoi continuare? (s -
n)"
4015 LET n$=INKEY$: IF n$<>"S" A
ND n$<>"N" THEN GO TO 4015
4020 IF n$="S" THEN POKE 23674,T
: POKE 23673,Y: POKE 23672,U: GO
TO 480
4030 CLS : PRINT TAB 14;"O.K.":
GO TO 170
5991 REM
5992 REM
5993 REM cambia gli attributi
dello schermo per
simulare il cubo
5994 REM
5995 REM
6000 FOR b=1 TO 4: FOR c=1 TO 4:
POKE 65001,a(schermo,b,c): POKE
65003,8+4*(c-1): POKE 65004,3+4
*(b-1): RANDOMIZE USR 65000: NEX
T c: NEXT b: RETURN

```

## ZX SPECTRUM

quando si riesce a completare il cubo. Da notare come questi dati non vengano "emmessi" di seguito come un unico blocco, ma sono divisi in due blocchi, rispettivamente di 8 e di 43 suoni. Questo allo scopo di evitare di annoiare con la stessa musica.

Se volete una ulteriore precisazione sul perché della struttura dell'array B così macchinosa, diremo semplicemente che ciò che è destra per una faccia del cubo non lo è per certe altre, e la stessa cosa vale per sù, giù, sinistra. Perciò ne risultava che i dati che noi volevamo trasferire in alto, andavano a finire a destra, con conseguente falsamento dell'andamento della partita.

## Routine Assembly

La routine che proponiamo insieme al programma Cubo Magico consente di cambiare gli attributi di una parte dello schermo.

Date le coordinate x,y dell'angolo in alto a sinistra del rettangolo di dimensioni a,b e un numero c rappresentante l'attributo da ottenere, possiamo far sì che dalle coordinate x,y appaia una "finestra" di dimensioni a,b di "colore" c (colore inteso come attributo).

Nel programma i valori di a e b sono costanti (a=4,b=4) e rappresentano il 7° e l'8° dato nella linea di Data del caricatore BASIC.

I valori di x e y variano a seconda della riga e della colonna della faccia a cui si è giunti (si notino i cicli For-Next di linea 6000) e rappresentano il 4° e il 5° dato nella linea di Data del caricatore BASIC. Il valore di c varia a seconda del valore di A (schermo,riga,colonna) e rappresenta il 2° valore nella linea di Data del caricatore BASIC.

Per utilizzare la routine al di fuori di questo programma bisogna dunque fare:

```
POKE ORG + 1,c
POKE ORG + 3,x
POKE ORG + 4,y
POKE ORG + 6,a
POKE ORG + 7,b
```

dove ORG = inizio routine. Ricordarsi di dare Clear ORG-1.

La routine va in esecuzione con Randomize Usr ORG, Let variabile = Usr, ORG, Print Usr ORG.

## Suddivisione in blocchi

**0-199** - Inizializzazione variabili. In particolare uso delle istruzioni Read, Data, Restore per riempire l'array B.

**200-350** - Parte grafica. Viene disegnato lo scheletro del cubo, il piccolo cubo tridimensionale in basso a sinistra, vengono stampate le lettere e le frecce rispettive.

**350-360** - Input del livello (tramite la In-key), ancora parte grafica che mostra il progressivo mischiamento sullo schermo.

**400-500** - Viene suonata un'allegra musicchetta in attesa che venga premuto Enter. La variabile "ma" è di servizio e serve per evitare un altro Goto inutile.

**500-600** - Controlla il tasto premuto. Salta alla relativa subroutine fornendo come parametri Numero, Verso. Se è premuto Enter salta a 3000.

**1000** - Routine che fa scorrere i dati verso l'alto o verso il basso.

**1100** - Routine che fa scorrere i dati verso destra o verso sinistra. Queste due ultime routine sono molto simili; si differenziano per un uso differente di Numero.

**2000** - Routine per cambiare lo schermo. È chiamata dalle linee 500-600.

**3000** - Routine per controllare se si è completato il cubo. In caso negativo salta a 4000.

**3200** - Congratulazioni per il successo del gioco.

**4000** - Viene chiamata se non si è completato il cubo. Ritorna a 500 se si vuole continuare. Ritorna all'inizio se si vuole ricominciare.

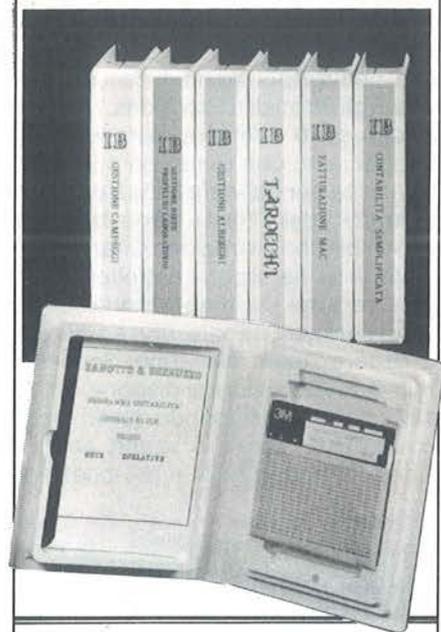
**6000** - Chiamata dalle routine 1000, 1100 e 2000, lancia la routine in linguaggio macchina, effettuando le opportune Poke. La routine è allocata a 65000, ma è completamente rilocabile. ■



INFORMATICA  
BIELLA

RIVENDITORE AUTORIZZATO

apple computer inc.



## Software

Contabilità generale 80CL Prodos  
Contabilità semplificata multaziendale  
Gestione Parrocchie  
Gestione Alberghi  
Parcellazione studi legali  
Fatturazione su MAC

## Hardware

Interfacce per Olivetti  
ET 121 / 201 / 221 / 111  
Interfacce per Adler  
G 8008 SE / 1005 / 1010 / 1030



INFORMATICA  
BIELLA

VIA ROMA 11  
13051 BIELLA  
TEL. 015 - 29.875  
24.181

**N**el n. 14 di **Personal Software** - Gennaio 1984 - apparve un articolo intitolato *Grafici ad alta risoluzione con il TI99/4A*. Seguirono molte telefonate di lettori che non riuscivano a far "girare" il programma che, nonostante tutto, funzionava benissimo. Speriamo che tutti, prima o poi, siano riusciti ad individuare gli errori di trascrizione. I controlli, a volte, sono veramente estenuanti. Supponiamo che ormai sia nota a tutti la tecnica per ottenere l'alta risoluzione. I caratteri del TI99/4A sono facilmente ridefinibili con l'istruzione Char; così, almeno in teoria, si può controllare il video pixel per pixel. Diciamo in teoria perché il set di caratteri, da solo, non è sufficiente a coprire l'intero schermo. Secondo il tipo di BASIC usato, si dispone di 111 o 127 caratteri, escludendo quello di codice ASCII 32 che è lo space, mentre il video possiede  $32 \times 24 = 768$  locazioni. È

un peccato che nel TI BASIC manchino le istruzioni per gestire la grafica in bitmap, tanto più se si considera che i nuovi computer dello standard MSX usano lo stesso microprocessore della Texas Instruments per il controllo del video e possiedono tutte le più comuni istruzioni grafiche, come Pset, Line, Cir-

cle, Paint, ecc. Nel precedente programma di Gennaio, solo apparentemente il grafico di una funzione occupava l'intero video; in realtà la maggior parte della superficie era costituita da Space. Nel caso di un grafico più "denso", utilizzati i caratteri disponibili, l'elaborazione si sarebbe in-

# Hi-Res

## Un programma per disegnare e stampare in alta risoluzione

di Sergio Borsani

**Listato 1 - Il programma HI-Res: ogni figura presenta un listato che indica il sottoprogramma utente da inserire nel blocco principale.**

```

100 REM *****
110 REM *
120 REM * HI-RES *
130 REM *
140 REM *****
150 REM *
160 REM * COPYRIGHT 1984 *
170 REM *
180 REM * BY S.BORSANI *
190 REM *
200 REM *****
210 REM *
220 REM * TI-99/4A *
230 REM *
240 REM * VERSIONE *
250 REM *
260 REM * EXTENDED BASIC *
270 REM *
280 REM *****
290 GOTO 390
300 CALL CLEAR :: CALL CHAR :: CALL HCHAR
R :: CALL VCHAR :: DIM S$(15,4),E$(15,4)
,M$(100)
310 DATA 8,4,2,1,9,5,3,1,A,6,2,3,B,7,3,3
,C,4,6,5,D,5,7,5,E,6,6,7,F,7,7,7
320 DATA 8,C,A,9,9,D,B,9,A,E,A,B,B,F,B,B
,C,C,E,D,D,D,F,D,E,E,E,F,F,F,F,F
330 DATA 0,0,0,0,1,1,1,0,2,2,0,2,3,3,1,2
,4,0,4,4,5,1,5,4,6,2,4,6,7,3,5,6
340 DATA 0,8,8,8,1,9,9,8,2,A,8,A,3,B,9,A
,4,B,C,C,5,9,D,C,6,A,C,E,7,B,D,E

```

```

350 J,K,FINE,N,CTR,MODD,X,Y,X1,Y1,S,XK,Y
K,Q,CO,RO,NC,LN,BI,PO,POX,A,NT,T,NUMCAR=
LOOP
360 NAME$,RECORD$,SC$,W$,A1$,DN$,FN$=" "
:: CALL CHARSET :: CALL COLOR :: CALL K
EY :: CALL GCHAR :: CALL CHARPAT
370 CALL LOCATE :: CALL SPRITE :: CALL D
ELSPRITE
380 !@P-
390 FOR J=0 TO 15 :: FOR K=1 TO 4 :: REA
D S$(J,K):: NEXT K :: NEXT J
400 FOR J=0 TO 15 :: FOR K=1 TO 4 :: REA
D E$(J,K):: NEXT K :: NEXT J :: GOTO 440
410 CALL CHAR(33,"00000000000000FF"):: C
ALL CHAR(34,"0101010101010101")
420 CALL CHAR(35,"8080808080808080"):: C
ALL CHAR(36,"FF"):: CALL CHAR(37,"3C4299
A1A199423C")
430 CALL CHAR(38,"FF818181818181FF"):: C
ALL CHAR(39,"8060703B1C0E0702"):: RETURN
440 CALL CHAR(140,"0042241818244200")::
CALL CHAR(141,"007C42427C4442")
450 CALL CHAR(142,"00183C7E181818"):: CA
LL CHAR(143,"001818187E3C1800")
460 FOR J=1 TO 100 :: M$(J)=RPT$("0",16)
:: NEXT J :: FINE=17
470 CALL CLEAR :: CALL CHARSET :: GOSUB
410 :: FOR J=2 TO 14 :: CALL COLOR(J,2,1
):: NEXT J
480 DISPLAY AT(2,7):"H I - R E S" :
: DISPLAY AT(4,2):"COPYRIGHT ";CHR$(37);
"1984 S. BORSANI"
490 CALL HCHAR(1,3,36,28):: CALL VCHAR(1
,2,34,5):: CALL VCHAR(1,31,35,5):: CALL
HCHAR(5,3,33,28)

```

terrotta. Un modo per ovviare a questo inconveniente consiste nel definire una finestra grafica tale da essere completamente occupata da caratteri ridefinibili, potendo così disporre a piacimento di tutti i pixel in essa contenuti.

Il programma presentato utilizza, appunto, una finestra grafica di 10 x 10 caratteri ovvero di 80 x 80 pixel. In essa si creano immagini o con l'aiuto della tastiera o con un sottoprogramma scritto dall'utente.

Si possono anche memorizzare le immagini in un file residente su nastro o su disco, richiamare in memoria vecchi disegni per eventuali modifiche ed infine stampare il disegno su carta. Il programma, insomma, si comporta come un semplice editor di immagini. La configurazione minima, richiede solo il BASIC esteso; la stampante non è indispensabile, in sua assenza ci si può accontentare dell'output sul video.

### Come si utilizza il programma

Il menu consiste nelle seguenti opzioni: Load, Draw, Save, Print, Erase e Files. Come suggeriscono i termini stessi, la prima carica in memoria un file nel quale è memorizzato un disegno precedentemente registrato con l'opzione Save; la seconda fa apparire la finestra grafica sulla quale viene realizzato il disegno; la terza registra un file contenente i pattern dei caratteri che formano il disegno; la quarta produce una copia su carta; la quinta azzerava la matrice adoperata per memorizzare il disegno; la sesta, infine, funzionante solo con un sistema a dischi, mostra i nomi dei file presenti sul dischetto.

Per eseguire un disegno, si seleziona l'opzione 2 del menu. Appare la finestra grafica, un cursore a forma di matita e quattro simboli nella parte bassa del video. Il loro significato, nell'ordine, è:

Pen Down (la penna scrive), Pen Up (la penna si sposta senza scrivere), Erase (la penna cancella), Return (si torna al menu). Il simbolo attivo è racchiuso da una cornice rossa ed inizialmente tale sorte tocca a Pen Up. Se esiste un sottoprogramma scritto dall'utente, questo viene eseguito subito, prima di poter disegnare con la tastiera. Si parlerà più avanti di questa possibilità. Per iniziare il disegno bisogna conoscere alcuni comandi fondamentali. Il cursore grafico a forma di matita si può trovare all'interno o all'esterno della finestra; il movimento nei due sensi avviene alla pressione del tasto Enter. Quando il cursore si trova all'esterno della finestra i tasti con le frecce <S> e <D> attivano una funzione operativa spostando la cornice sul simbolo corrispondente, quando invece il cursore si trova all'interno, otto tasti provvedono al suo movimento nelle direzioni principali e lungo le diagonali.

```

500 DISPLAY AT(8,4):"1 LOAD" :: DISPLAY
AT(10,4):"2 DRAW" :: DISPLAY AT(12,4):"3
SAVE"
510 DISPLAY AT(14,4):"4 PRINT" :: DISPLA
Y AT(16,4):"5 ERASE" :: DISPLAY AT(18,4)
:"6 FILES"
520 DISPLAY AT(20,10):"SCELTA?" :: ACCEP
T AT(20,18).SIZE(1)VALIDATE("123456"):SC$
530 ON ERROR 1350 :: N=VAL(SC$):: CALL H
CHAR(6+2*N,4,38):: ON N GOTO 540,580,107
0,1110,1170,1200
540 GOSUB 1420 :: IF NAME$="CS1" THEN CA
LL CLEAR
550 OPEN #1:NAME$,SEQUENTIAL,INPUT ,DISP
LAY ,FIXED 80
560 FOR J=1 TO 20 :: INPUT #1:RECORD$ ::
FOR K=1 TO 5 :: M$((J-1)*5+K)=SEG$(RECO
RD$, (K-1)*16+1,16):: NEXT K :: NEXT J
570 CLOSE #1 :: IF NAME$="CS1" THEN 470
ELSE GOSUB 1410 :: GOTO 520
580 CALL CLEAR :: GOSUB 1490 :: FOR J=2
TO 14 :: CALL COLOR(J,2,16):: NEXT J ::
CTR=-1
590 FOR J=8 TO 17 :: FOR K=12 TO 21 :: C
TR=CTR+1 :: CALL HCHAR(J,K,40+CTR):: NEX
T K :: NEXT J :: CALL UTENTE
600 MOD0=2 :: X=127 :: Y=97 :: X1=113 ::
Y1=169 :: GOSUB 1360 :: GOTO 860
610 CALL KEY(O,K,S):: IF S=0 THEN 610
620 IF K=13 THEN 860
630 IF K=69 THEN XK=0 :: YK=1 :: GOTO 71
0
640 IF K=82 THEN XK,YK=1 :: GOTO 710
650 IF K=68 THEN XK=1 :: YK=0 :: GOTO 71
0

```

```

660 IF K=67 THEN XK=1 :: YK=-1 :: GOTO 7
10
670 IF K=88 THEN XK=0 :: YK=-1 :: GOTO 7
10
680 IF K=90 THEN XK,YK=-1 :: GOTO 710
690 IF K=83 THEN XK=-1 :: YK=0 :: GOTO 7
10
700 IF K=87 THEN XK=-1 :: YK=1 ELSE 610
710 Q=Y-YK :: IF Q>56 AND Q<137 THEN Y=Q
720 Q=X+XK :: IF Q>88 AND Q<167 THEN X=Q
730 IF MOD0=2 THEN 840
740 CO=INT((X-1)/8)+1 :: RO=INT((Y-1)/8)
+1
750 CALL GCHAR(RO,CO,NC):: CALL CHARPAT(
NC,A1$):: LN=Y-INT((RO-1)*8):: BI=X-INT(
(X-1)/4)*4 :: PO=INT((X-1)/4)+1
760 IF PO/2=INT(PO/2) THEN POX=LN*2 ELSE
POX=LN*2-1
770 W$=SEG$(A1$,POX,1):: IF W$="F" AND M
OD0=1 THEN 840
780 IF W$="O" AND MOD0=3 THEN 840
790 A=ASC(W$):: IF A>60 THEN NT=A-55 ELS
E NT=A-48
800 IF MOD0=1 THEN W$=S$(NT,BI)ELSE W$=E
$(NT,BI)
810 IF POX=1 THEN A1$=W$&SEG$(A1$,2,15):
: GOTO 840
820 IF POX=16 THEN A1$=SEG$(A1$,1,15)&W$
:: GOTO 840
830 A1$=SEG$(A1$,1,POX-1)&W$&SEG$(A1$,PO
X+1,16-POX)
840 CALL LOCATE(#1,Y,X):: IF MOD0=2 THEN
610
850 CALL CHAR(NC,A1$):: GOTO 610
860 CALL LOCATE(#1,145,140):: FOR T=1 TO

```

Come è ovvio, in queste applicazioni i tasti sono quelli con le frecce ed i loro vicini: E=alto, X=basso, S=sinistra, D=destra, R=alto-destra, C=basso-destra, Z=basso-sinistra, W=alto-sinistra. È stato previsto l'uso esclusivo della tastiera poiché i joystick, che teoricamente sembravano possedere le carte migliori, all'atto pratico si sono rivelati poco sensibili agli spostamenti in diagonale.

Un disegno, generalmente richiede un frequente cambio di funzioni, ma con un po' di pratica non è difficile controllare il movimento del cursore grafico, farlo uscire dalla finestra con il tasto Enter, cambiare la funzione e tornare nello stesso punto con il cursore per riprendere il disegno.

Quando il lavoro è terminato, per tornare al menu, bisogna attivare il simbolo R sulla destra e attendere brevemente perché il computer memorizzi le stringhe esadecimali che definiscono i caratteri, prima che essi assumano nuovamente il loro aspetto abituale. Dopo questa fase appare una linea alla base della finestra; la linea può essere spostata con i tasti Fctn E e Fctn X rispettivamente verso l'alto e verso il basso. Lo scopo è quello di "tagliare" la finestra riducendo in tal modo la superficie da stampa-

re. Chi invece desidera stampare tutta la finestra o non intende affatto eseguire la stampa, premerà semplicemente il tasto Enter, tornando così al menu principale. La precedente operazione per limitare la superficie da stampare si rende necessaria per la relativa lentezza con la quale viene eseguita la copia su carta. Se il disegno non occupa tutta la finestra e si è avuta l'accortezza di realizzarlo nella parte alta, un "taglio" farà risparmiare in seguito del tempo prezioso impedendo la stampa di inutili caratteri vuoti.

### Il sottoprogramma utente

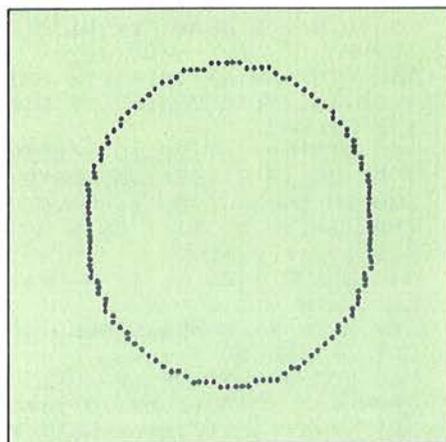
Una parte interessante del programma è certamente quella rivolta al sottoprogramma che l'utente può scrivere e inserire nel listato, dopo la linea 2100, non dimenticando di terminare con l'istruzione Subend. Per facilitare la programmazione sono state create tre nuove istruzioni grafiche con altrettanti sottoprogrammi BASIC. Esse sono: Pset per segnare un punto; Line per tracciare una linea e Circle per tracciare cerchi ed ellissi. Queste nuove istruzioni, come tutti i sottoprogrammi, vengono chiamate con delle Call rispettando una precisa sintassi.

**Call Pset (x,y)** attiva un pixel della finestra grafica colorandolo di nero. L'origine del sistema di riferimento è posto nell'angolo in basso a sinistra ed i punti sono numerati dallo 0 al 79. I valori esterni a tale intervallo non provocano alcun errore, ma vengono semplicemente ignorati.

**Call Line (x1, y1, x2, y2)** traccia una linea dal punto di coordinate x1, y1 al punto di coordinate x2, y2. Se entrambi i punti cadono all'esterno della finestra l'istruzione viene ignorata, se invece uno solo degli estremi è esterno, allora viene tracciato solo il segmento che

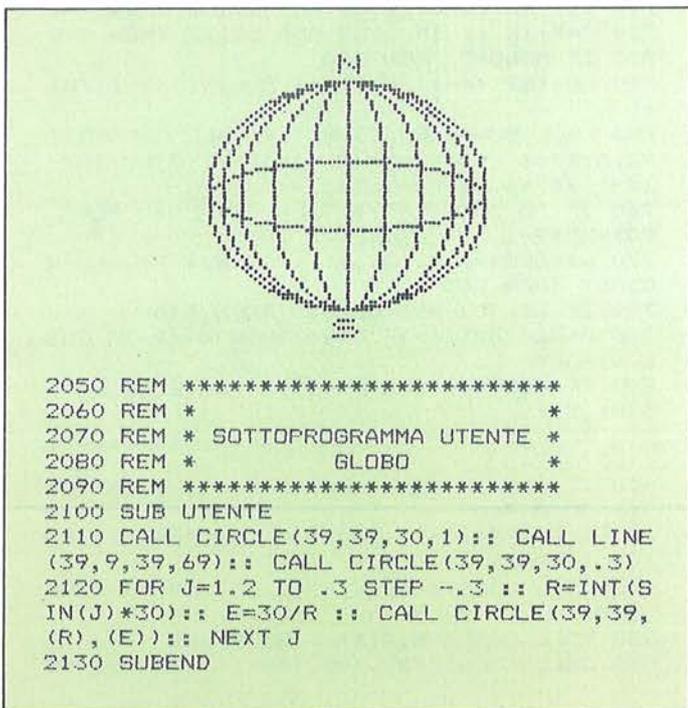
**Figura 2 - A causa della diversa distanza dei punti che formano l'immagine nelle due direzioni, verticale ed orizzontale, una circonferenza, calcolata matematicamente, appare come un'ellisse. Una diversa deformazione si ha in fase di stampa per la diversa distanza degli aghi della testina della stampante.**

**La figura mostra la riproduzione di un cerchio che appariva tale sul video, ottenuto con un'eccentricità uguale a 1,38.**



rientra nella finestra grafica.

**Call Circle (x0, y0, r, e)** traccia un cerchio di centro x0, y0, di raggio r e di eccentricità e. Anche qui esiste la funzione di "clipping" per tracciare ugualmente la curva qualora una parte dovesse cadere all'esterno della finestra. Per quanto riguarda l'eccentricità, da un punto di vista geometrico, il valore 1 corrisponde ad una circonferenza, valori diversi da 1 ad ellissi. In realtà i punti che formano il video non sono equidistanti nelle due direzioni, orizzontale e verticale, con il risultato che un'eccentricità uguale a 1 fa apparire un'ellisse. Un



**Figura 1 - Immagine di un globo creata con un sottoprogramma che utilizza le nuove istruzioni Line e Circle. Le lettere N e S sono state aggiunte controllando con la tastiera il cursore grafico.**

valore di 1,39 dà buoni risultati sullo schermo, facendo apparire un cerchio; purtroppo in fase di stampa, a causa della diversa conformazione della matrice dei punti, si ottiene nuovamente un'ellisse, anche se non molto eccentrica. In ogni caso, mancando un valore di default, bisogna indicare il valore che si ritiene più opportuno.

La figura 2 mostra appunto un cerchio che appariva veramente tale sul video; durante la stampa ha subito una leggera deformazione.

Le figure 3 e 4 mostrano immagini ottenute con altrettanti sottoprogrammi u-

sando istruzioni Line e Circle, completate poi tramite la tastiera.

### La stampa

Per quanto riguarda la stampa, è necessario conoscere alcune caratteristiche della stampante. Al momento dell'acquisto la testina di stampa è regolata dalla fabbrica in modo da lavorare con 7 aghi di stampa. Tuttavia, per poter sfruttare pienamente le sue capacità grafiche è necessario tarare in posizione Off l'interruttore dip SW2-1 posto sul quadro dell'interfaccia seriale. L'opera-

zione è semplice, e per eseguirla basta seguire le istruzioni contenute nel manuale. Se non sono state apportate altre modifiche alla taratura degli interruttori dip della stampante, tutto funzionerà regolarmente. Spesso, tuttavia, viene cambiata la velocità di trasmissione dati che la Casa ha fissato a 300 baud. Con velocità diverse l'utente deve modificare la linea 1880 del programma, contenente l'istruzione Open #4: "RS232.CR.DA = 8", per adattarla alle proprie necessità.

### Come si ottiene l'indirizzamento indipendente dei punti?

Prima di terminare, merita una breve spiegazione il modo in cui viene messo in On un pixel della finestra grafica.

I punti del video non sono indirizzabili indipendentemente poiché la memoria è organizzata in modo da contenere, da un lato, i numeri dei caratteri presenti in ciascuna locazione dello schermo, dall'altro la forma dei caratteri stessi, codificata, a livello utente, con stringhe di 16 caratteri esadecimali. In pratica, per mettere in On un pixel, bisogna ridefinire il carattere al quale appartiene. Se osserviamo le linee del programma a partire dalla numero 740, per prima cosa vengono trovati i numeri di colonna e di riga (CO e RO) della locazione dove si trova il cursore. Il numero di codice ASCII del carattere (NC) viene ricavato con l'istruzione Call Gchar mentre la successiva Call Chrpat resituisce il pattern del carattere, cioè la stringa di caratteri esadecimali (A1\$) ognuno dei quali controlla un semibyte in memoria,

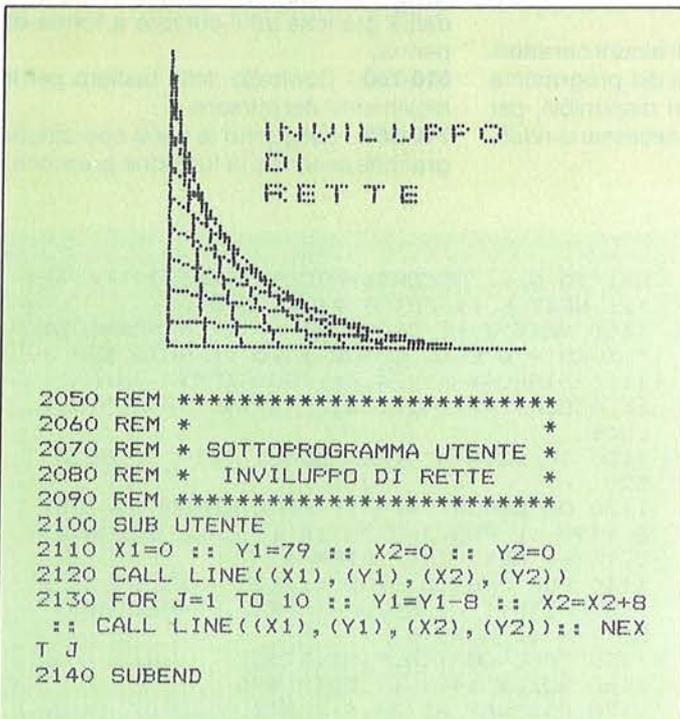


Figura 3 - Inviluppo di rette ottenuto con un ciclo For-Next nel quale viene chiamato il sottoprogramma Sub Line. Il testo è stato inserito controllando il cursore grafico con la tastiera. Infatti, creando una finestra in bit map non si dispone più del normale set di caratteri alfanumerici.

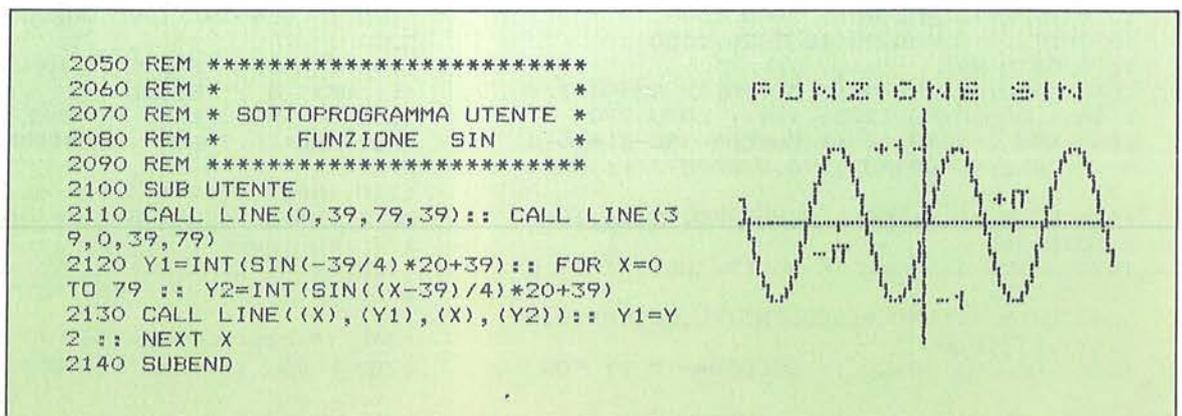


Figura 4 - Esempio di grafico di funzione.

ovvero una sequenza di 4 pixel. Individuato il semibyte, si tratta di modificare un bit lasciando inalterati gli altri tre. A questo punto si presenta un'alternativa: passare dalla notazione esadecimale a quella decimale, usare gli operatori booleani e poi tornare nuovamente all'esadecimale, oppure creare delle tabelle di composizione (una sorta di tavole pitagoriche) per operare direttamente con caratteri esadecimale. Nel programma è stata adottata questa seconda soluzione. Le tabelle sono contenute nelle matrici S\$( ) e E\$( ) e servono rispettivamente per mettere in On o in Off un bit e conseguentemente un pixel del video. Individuata la posizione del carattere esadecimale nella stringa (POX), il suo valore decimale corrisponde ad una riga della tabella, mentre il numero del bit (BI) all'interno del semibyte, corrisponde ad una colonna della tabella. L'elemento S\$(NT,BI) restituisce il nuo-

vo carattere esadecimale che codifica il semibyte modificato. Ricostruita infine la stringa che costituisce il pattern del carattere, apparirà un nuovo punto sullo schermo.

### Descrizione del listato

**300-380** - Elenco delle variabili per il prescan. È questa una tecnica per abbreviare il tempo d'attesa all'inizio del programma.

**390-400** - Caricamento delle tabelle per effettuare operazioni logiche, in esadecimale, tra bit e semibyte corrispondenti. In pratica tali tabelle serviranno per porre in On o in Off un pixel della finestra grafica.

**410-460** - Definizione di alcuni caratteri. Poiché la parte grafica del programma assorbe tutti i caratteri disponibili, per poter riporre il menu, è necessario ridefinire alcuni caratteri.

**470-530** - Menu. La linea 470 ristabilisce il normale set di caratteri alfanumerici.

**540-570** - Eseguono il comando Load. La Gosub di linea 540 chiede il nome del file.

**580-590** - In esecuzione del comando Draw, appare la finestra grafica e vengono ridefiniti i caratteri (subroutine 1490) per far apparire il disegno attualmente in memoria e conservato nel vettore M\$( ). Prima di proseguire viene lanciato il sottoprogramma Sub Utente. Se questo non contiene istruzioni si passa all'esecuzione del disegno sotto il controllo della tastiera.

**600** - Chiama una subroutine per far apparire i simboli per selezionare le modalità grafiche ed il cursore a forma di penna.

**610-700** - Controllo della tastiera per il movimento del cursore.

**710-850** - Eseguono le varie operazioni grafiche secondo la funzione prescelta.

### Seguito programma HI-Res

```

50 :: NEXT T
870 CALL KEY(O,K,S):: IF S=0 THEN 870
880 IF K=13 THEN 950
890 IF K=68 AND MOD0<4 THEN MOD0=MOD0+1
:: GOTO 910
900 IF K=83 AND MOD0>1 THEN MOD0=MOD0-1
:: GOTO 910 ELSE 870
910 IF MOD0=1 THEN X1=97 :: GOTO 940
920 IF MOD0=2 THEN X1=113 :: GOTO 940
930 IF MOD0=3 THEN X1=129 ELSE X1=153
940 CALL LOCATE(#2,Y1,X1):: GOTO 860
950 IF MOD0<>4 THEN 730
960 FOR J=1 TO 100 :: CALL CHARPAT(J+39,
A1$):: M$(J)=A1$ :: NEXT J
970 CALL DELSPRITE(ALL)
980 CALL HCHAR(18,12,36,10):: RO=18
990 CALL KEY(O,K,S):: IF S=0 THEN 990
1000 IF K=13 THEN FINE=RO-1 :: GOTO 470
1010 IF K=11 AND RO>9 THEN 1040
1020 IF K=10 AND RO<18 THEN 1050
1030 GOTO 990
1040 CALL HCHAR(RO,12,32,10):: RO=RO-1 :
: CALL HCHAR(RO,12,36,10):: GOTO 990
1050 FOR J=1 TO 10 :: NUMCAR=(RO-8)*10+3
9 :: CALL HCHAR(RO,11+J,NUMCAR+J):: NEXT
J
1060 RO=RO+1 :: CALL HCHAR(RO,12,36,10):
: GOTO 990
1070 GOSUB 1420 :: IF NAME$="CS1" THEN C
ALL CLEAR
1080 OPEN #1:NAME$,SEQUENTIAL,OUTPUT,DIS
PLAY ,FIXED 80
1090 FOR J=1 TO 20 :: RECORD$="" :: FOR

```

```

K=1 TO 5 :: RECORD$=RECORD$&M$((J-1)*5+K
):: NEXT K :: PRINT #1:RECORD$
1100 NEXT J :: CLOSE #1 :: IF NAME$="CS1
" THEN 470 ELSE GOSUB 1410 :: GOTO 520
1110 DISPLAY AT(24,6):"CONFERMI? (Y/N)"
:: ACCEPT AT(24,22)SIZE(1)VALIDATE("YN")
:SC$
1120 IF SC$="N" THEN GOSUB 1410 :: GOTO
520
1130 ON ERROR 1440 :: CALL CLEAR :: GOSU
B 1490 :: FOR J=2 TO 14 :: CALL COLOR(J,
2,16):: NEXT J :: CTR=-1
1140 FOR J=8 TO FINE :: FOR K=12 TO 21 :
: CTR=CTR+1 :: CALL HCHAR(J,K,40+CTR)::
NEXT K :: NEXT J
1150 CALL COPY(8,FINE,12,21)
1160 GOSUB 1410 :: GOTO 470
1170 DISPLAY AT(24,6):"SEI SICURO? (Y/N)
" :: ACCEPT AT(24,24)SIZE(1)VALIDATE("YN
"):SC$
1180 IF SC$="N" THEN GOSUB 1410 :: GOTO
520
1190 FOR J=1 TO 100 :: M$(J)=RPT$("0",16
):: NEXT J :: GOSUB 1410 :: GOTO 520
1200 ON ERROR 1470 :: CALL CLEAR :: OPEN
#3:"DSK1.",INPUT ,RELATIVE,INTERNAL ::
CTR=0
1210 INPUT #3:DN$,J,J,K :: PRINT "DISKNA
ME=";DN$:"AVAILABLE=";K;"USED";J-K
1220 FOR LOOP=1 TO 127 :: INPUT #3:FN$,Y
,J,K :: IF LEN(FN$)=0 THEN 1280
1230 IF ABS(Y)=2 THEN PRINT FN$,:: CTR=C
TR+1
1240 IF CTR<40 THEN 1270
1250 PRINT : "PRESS <ENTER> TO CONTINUE"

```

## TEXAS TI99/4A

In particolare, le linee 740-760 individuano il carattere nel quale si trova il cursore ed il semibyte nel quale si trova il bit da modificare.

**860-940** - Controllo della tastiera per la selezione di una funzione grafica (Pen Up, Pen Down, ecc.).

**950-960** - Il disegno viene memorizzato nel vettore M\$.

**970-1060** - Si permette il "taglio" della finestra grafica in modo da impedire la stampa di Space con un'inutile perdita di tempo.

**1070-1100** - Esecuzione del comando Save con il quale viene memorizzato su nastro o su disco il vettore M\$ contenente i pattern dei caratteri che, in precedenza, formavano il disegno.

**1110-1160** - La sezione effettua l'hard copy su carta dell'immagine presente nella finestra grafica. A tal fine viene chiamato il sottoprogramma Sub Copy. Nella Call, tra parentesi, vengono tra-

smessi i numeri di riga e di colonna che delimitano la finestra grafica.

**1170-1190** - Azzeramento degli elementi del vettore M\$ e conseguentemente del disegno presente in memoria.

**1200-1340** - Lettura della directory per la lista dei file presenti sul dischetto ed eventuale cancellazione di un file.

#### Subroutine

**1350** - Subroutine legata ad una On Error. Rimanda il programma al menu.

**1360-1400** - Fa apparire i simboli per selezionare le modalità grafiche ed il cursore a forma di penna.

**1410** - Cancella la parte inferiore del video.

**1420-1430** - Accetta il nome del file contenente un'immagine da memorizzare.

**1440-1460** - Subroutine per la gestione degli errori.

**1470-1480** - Messaggio conseguente all'uso del comando Files con il registra-

tore a cassette. Tale opzione è consentita solo con il sistema a dischi.

**1490-1500** - Ridefinisce i caratteri con i pattern memorizzati nel vettore M\$.

#### Sottoprogrammi

**1530-1670** - Sub Pset. Pone in On un pixel del video dopo aver individuato il carattere ed il semibyte corrispondenti.

**1680-1780** - Sub Line. Traccia una linea utilizzando il precedente sottoprogramma.

**1790-1860** - Sub Circle. Traccia un cerchio.

**1870-2040** - Sub Copy. Esegue una copia su carta del disegno presente nella finestra grafica. La linea 1880 contiene la Open per l'interfaccia seriale. Si presuppone che la taratura degli interruttori dip della stampante sia quella originale ad eccezione di quello che attiva parole di 8 bit (vedi articolo).

**2050-2110** - Sub Utente. Nel listato que-

#### Seguito programma HI-Res

```

1260 CALL KEY(O,K,S):: IF S=0 THEN 1260
ELSE PRINT :: CTR=0
1270 NEXT LOOP
1280 CLOSE #3 :: PRINT "DETETE A FILE?
(Y/N) N" :: ACCEPT AT(23,22)SIZE(-1)VALI
DATE("YN"):SC#
1290 IF SC#="N" THEN 470
1300 PRINT :: PRINT :: PRINT
1310 GOSUB 1420 :: DISPLAY AT(24,6):"SEI
SICURO? (Y,N) N"
1320 ACCEPT AT(24,24)SIZE(-1)VALIDATE("Y
N"):SC# :: IF SC#="N" THEN 470
1330 OPEN #2:NAME$,SEQUENTIAL,INPUT ,DIS
PLAY ,FIXED 20 :: CLOSE #2:DELETE :: GOT
O 470
1340 END
1350 CALL DELSPRITE(ALL):: RETURN 470
1360 CALL HCHAR(22,13,143):: CALL HCHAR(
22,15,142):: CALL HCHAR(22,17,140):: CAL
L HCHAR(22,20,141)
1370 CALL HCHAR(21,12,36,10):: CALL HCHA
R(23,12,33,10)
1380 CALL VCHAR(21,11,34,3):: CALL VCHAR
(21,22,35,3)
1390 CALL SPRITE(#1,39,7,145,140,0,0,#2,
38,9,Y1,X1,0,0)
1400 RETURN
1410 CALL HCHAR(23,3,32,60):: CALL VCHAR
(8,4,32,12):: RETURN
1420 ON ERROR 1440 :: DISPLAY AT(23,1):"
FILE NAME?" :: ACCEPT AT(23,12)SIZE(15):
NAME#
1430 RETURN

```

```

1440 DISPLAY AT(24,1):"* ERROR * PRESS <
ENTER>"
1450 CALL KEY(O,K,S):: IF S=0 THEN 1450
1460 IF K=13 THEN GOSUB 1410 :: RETURN 5
20 ELSE 1450
1470 PRINT "ONLY FOR DISK SYSTEM!": " " :
PRESS <ENTER> TO CONTINUE"
1480 CALL KEY(O,K,S):: IF S=0 OR K<>13 T
HEN 1480 ELSE RETURN 470
1490 FOR J=40 TO 139 :: CALL CHAR(J,M$(J
-39)):: NEXT J
1500 RETURN
1510 !@P+
1520 END
1530 SUB PSET(X,Y)
1540 X=INT(X+.5):: Y=INT(Y+.5)
1550 IF X<0 OR X>79 THEN SUBEXIT
1560 IF Y<0 OR Y>79 THEN SUBEXIT
1570 RO=17-INT(Y/8):: CO=INT(X/8)+12 ::
CALL GCHAR(RO,CO,NC):: CALL CHARPAT(NC,P
#)
1580 LN=(INT(Y/8)+1)*8-Y :: BI=X-INT(X/4
)*4 :: PO=INT(X/4)+1
1590 IF PO/2=INT(PO/2) THEN POX=LN*2 ELSE
POX=LN*2-1
1600 W#=SEG$(P$,POX,1):: IF W#="F" THEN
SUBEXIT
1610 A=ASC(W#):: IF A>60 THEN DEC=A-55 E
LSE DEC=A-48
1620 DEC=DEC OR 2^(3-BI):: IF DEC<10 THE
N W#=STR$(DEC)ELSE W#=CHR$(DEC+55)
1630 IF POX=1 THEN P#=W#&SEG$(P$,2,15)::
GOTO 1660
1640 IF POX=16 THEN P#=SEG$(P$,1,15)&W#
:: GOTO 1660

```

## TEXAS TI99/4A

sto sottoprogramma non contiene alcuna istruzione; per il buon funzionamento, la sua presenza è ugualmente necessaria.

### Principali variabili del programma

**SS()** - Tabella a due dimensioni per mettere in On un bit di un semibyte espresso nella notazione esadecimale.

**ES()** - Tabella simile alla precedente usata per mettere in Off un bit.

**MS()** - Vettore usato per memorizzare i pattern dei caratteri quando questi devono essere resettati.

**Name\$** - Nome del dispositivo usato come memoria di massa (nastro o disco).

**Record\$** - Record del file costituito dal pattern di 5 caratteri. Si ricorda che un pattern viene codificato con una stringa di 16 caratteri esadecimale.

**SC\$** - Scelta dell'utente al menu.

**W\$, A1\$** - Variabili di lavoro usate per ottenere la stringa di caratteri esadecimale che ridefiniscono un carattere dopo che è stato modificato mettendo in On un pixel.

**DN\$** - Variabile utilizzata per leggere il nome del dischetto dalla directory.

**FN\$** - Variabile usata per leggere i nomi dei file dalla directory.

**J,K** - Variabili di lavoro adoperate prevalentemente nei cicli.

**Fine** - Contiene il numero di riga del video che delimita la superficie da stampare.

**Modo** - Assume i valori 1, 2, 3 e 4, corrispondenti ai quattro simboli posti al di sotto della finestra grafica.

**X, Y, X1, Y1** - Coordinate di un punto all'interno della finestra grafica.

**XK, YK** - Assumono i valori -1, 0, +1 per indicare lo spostamento relativo del cur-

sore grafico.

**Q** - Variabile di comodo per il calcolo di una nuova coordinata.

**CO,RO** - Numero di colonna e numero di riga.

**NC** - Numero di codice ASCII di un carattere.

**LN** - Una delle otto linee di otto punti che formano un carattere nella quale si trova il cursore grafico.

**BI** - Bit, di un semibyte, che va posto in On.

**PO** - Indica la posizione del semibyte specificando se è di sinistra o di destra.

**POX** - Indica la posizione di un carattere esadecimale all'interno della stringa che definisce un carattere.

**NT** - Numero di un elemento delle tabelle contenute nelle matrici S\$ e E\$.

**Numcar** - Numero di un carattere presente nella finestra grafica. ■

#### Seguito programma HI-Res

```

1650 P$=SEG$(P$, 1, POX-1)&W$&SEG$(P$, POX+
1, 16-POX)
1660 CALL CHAR(NC, P$)
1670 SUBEND
1680 SUB LINE(X1, Y1, X2, Y2)
1690 IF X1=X2 THEN X=X1 :: GOTO 1760
1700 IF X1>X2 THEN Q=X1 :: X1=X2 :: X2=Q
:: Q=Y1 :: Y1=Y2 :: Y2=Q
1710 YO=Y1 :: CALL PSET((X1), (Y1))
1720 FOR X=X1 TO X2
1730 Y=INT((X-X1)*(Y2-Y1)/(X2-X1))+Y1 ::
W=INT(Y-YO):: IF ABS(W)>1 THEN Y=YO+SGN
(W):: FLAG=1
1740 CALL PSET((X), (Y)):: YO=Y :: IF FLA
G=1 THEN FLAG=0 :: GOTO 1730
1750 NEXT X :: SUBEXIT
1760 IF Y1>Y2 THEN Q=Y1 :: Y1=Y2 :: Y2=Q
1770 FOR Y=Y1 TO Y2 :: CALL PSET(X, Y)::
NEXT Y
1780 SUBEND
1790 SUB CIRCLE(X0, YO, R, E)
1800 Y1=YO+R*E :: CALL PSET((X0), (Y1))
1810 FOR X=X0 TO X0+R
1820 Y=YO+SQR(R*R-(X0-X)^2)*E :: W=INT(Y
-Y1):: IF ABS(W)>1 THEN Y=Y1+SGN(W):: FL
AG=1
1830 CALL PSET((X), (Y)):: CALL PSET(X, 2*
YO-Y):: CALL PSET((2*X0-X), (Y)):: CALL P
SET((2*X0-X), (2*YO-Y))
1840 Y1=Y :: IF FLAG=1 THEN FLAG=0 :: GO
TO 1820
1850 NEXT X
1860 SUBEND
1870 SUB COPY(R1, R2, C1, C2)

```

```

1880 OPEN #4:"RS232.CR.DA=8" :: PRINT #4
:CHR$(27);CHR$(65);CHR$(8);
1890 FOR R=R1 TO R2 :: FOR C=C1 TO C2 ::
CALL GCHAR(R, C, N):: CALL CHARPAT(N, W$)
1900 PRINT #4:CHR$(27);"K";CHR$(8);CHR$(
0);
1910 FOR K=3 TO 0 STEP -1 :: Q=0 :: FOR
E=1 TO 15 STEP 2
1920 D=ASC(SEG$(W$, E, 1)):: IF D<60 THEN
D=D-48 ELSE D=D-55
1930 D=D AND 2^K :: IF D THEN Q=Q+2^(7-I
NT(E/2))
1940 NEXT E :: PRINT #4:CHR$(Q);
1950 NEXT K
1960 FOR K=3 TO 0 STEP -1 :: Q=0 :: FOR
E=2 TO 16 STEP 2
1970 D=ASC(SEG$(W$, E, 1)):: IF D<60 THEN
D=D-48 ELSE D=D-55
1980 D=D AND 2^K :: IF D THEN Q=Q+2^(8-I
NT(E/2))
1990 NEXT E :: PRINT #4:CHR$(Q);
2000 NEXT K
2010 NEXT C :: PRINT #4:CHR$(13);CHR$(10
);
2020 NEXT R
2030 PRINT #4:CHR$(27);CHR$(65);CHR$(12)
;:: CLOSE #4
2040 SUBEND
2050 REM *****
2060 REM *
2070 REM * SOTTOPROGRAMMA UTENTE *
2080 REM *
2090 REM *****
2100 SUB UTENTE
2110 SUBEND

```

# FAI CRESCERE

# IL TUO TI99/4A CON LA

# NUOVA FORMIDABILE

# OFFERTA J.SOFT

Dopo il successo ottenuto dalla prima proposta, ecco a disposizione del tuo TI-99/4A un assortimento di altri splendidi programmi, tra i quali il richiestissimo EXTENDED BASIC; il MODULO ADVENTURE, che ti permetterà di vivere appassionanti avventure e l'EDITOR/ASSEMBLER, indispensabile per il definitivo salto di qualità del tuo computer.

Ordina subito, per essere sicuro di ricevere i programmi al più presto.



## EXTENDED BASIC

Cod. DTXNX01 - L. 270.000

Finalmente disponibile il linguaggio di programmazione compatibile con il BASIC TI-99/4A, con le seguenti aggiunte: righe multidichiarazioni, dichiarazioni IF-THEN-ELSE, accesso diretto allo schermo, sottoprogrammi BASIC con variabili locali, autoavviamento dei programmi su disco, facile controllo di fino a 28 sprite con concatenamento programmi BASIC, fusione codici, programmi protetti, dichiarazioni ON ERROR (Errore in atto) e funzioni logiche di Boole (AND, OR, NOT 1E, O, NO, ecc.).

Permette, con l'espansione della memoria, di programmare in linguaggio macchina.

## MODULO AVVENTURA

Cod. DTXRX01 - L. 70.000

Si tratta del modulo di comando, da inserire nella console, che funge da interprete del linguaggio delle cassette sottodescritte.

Uno strumento indispensabile per arricchire la tua software-teca di nuove entusiasmanti avventure.

Il modulo viene fornito con l'avventura "IL PIRATA".

Configurazione richiesta: registratore a cassetta.



### Elenco delle "Avventure" che girano esclusivamente col MODULO AVVENTURA

Nove programmi entusiasmanti, semplici da usare ma difficili da finire.

All'inizio di ogni avventura viene proposta la situazione in cui ci si trova e viene posta la domanda: COSA DEVO FARE? Utilizzando gli oltre 200 vocaboli previsti dal vocabolario del computer, darai gli ordini conseguenti, per trovarti in situazioni sempre diverse fino a realizzare lo scopo del gioco, ogni volta differente per ogni titolo. Avventure che stimolano la tua capacità di ragionare in modo logico e che richiedono ore, e anche settimane, per concludersi.

#### Mystery Fon House (La casa del Mistero)

Cod. DTXMX17 - L. 30.000

Prima di procedere all'esplorazione, devi meditare su come fare per entrare nella Casa del Mistero.

Una volta dentro, potrai vedere tutti gli angoli della casa, dove si nasconde un prezioso premio.

#### Pyramid of Doom (La Piramide del Destino)

Cod. DTXMX22 - L. 30.000

L'avventura inizia in un deserto, nei pressi di uno stagno dal quale fuoriesce un palo. Grazie alle tue ricerche, troverai una piramide solo di recente scoperta dalle sabbie mobili. Cercherai l'ingresso, raccoglierai tesori e cercherai di fuggire dalla piramide.

#### Strange Odyssey (Strana Odissea)

Cod. DTXMX24 - L. 30.000

Inizia quando ti rendi conto di essere "naufragato" su un minuscolo pianeta e quindi devi riparare l'astronave per poter tornare a casa.

Mentre cerchi i pezzi di ricambio sul pianeta, tenterai di scoprire i segreti di un'antica civiltà, raccogliendo i tesori in cui ti imbattevi lungo la strada.

#### Adventureland

Cod. DTXMX01 - L. 30.000

Ti trovi all'inizio della foresta di un mondo di fiaba. Esplorandolo, potrai scoprire 13 diversi tesori come pure il luogo particolare dove riporli.

#### Voodoo Castle (Castello del Voodoo)

Cod. DTXMX32 - L. 30.000

Sei all'interno di una cappella di fronte ad una bara chiusa. Esplorando il castello, cercherai di raccogliere le informazioni necessarie per liberare il Conte Cristo dalla terribile maledizione che i suoi nemici gli hanno fatto.

#### Ghost Town (La Città Fantasma)

Cod. DTXMX09 - L. 30.000

Tutte le città fantasma sono misteriose, ma questa ha un fascino particolare: è abitata da fantasmi veri ed è piena di tesori. Esplora tutti i vecchi edifici per vedere quanti ne riuscirai a trovare.

#### Savage Island Series (Serie dell'Isola Selvaggia)

Cod. DTXMX35 - L. 30.000

L'avventura si divide in due parti ed inizia ai margini di una giungla impenetrabile. Mentre esplori l'isola, incontrerai delle creature insolite. Dopo aver completato la prima parte, riceverai la parola d'ordine per iniziare la seconda parte, dove o diventerai il più grande eroe al mondo o andrai incontro a una fine rapida ed orribile.

#### The Count (Il conte)

Cod. DTXMX26 - L. 30.000

Ti svegli da un pisolino e ti trovi in un letto con un paletto da tenda in mano. Sta a te scoprire chi sei, cosa stai facendo in Transilvania e perché il postino ha consegnato un flacone di sangue!

#### The Golden Voyage

Cod. DTXMX10 - L. 30.000

Comincerai con l'individuare il palazzo reale nella città persiana. Nel palazzo incontrerai l'anziano re al quale rimangono solo tre giorni di vita, a meno che non riuscirai a ringiovanirlo. Portando soltanto una borsa d'oro, salperai in mare alla ricerca della mitica fontana della giovinezza.

I programmi seguenti sono invece moduli Solid State Software, pronti da inserire:

**Music Maker** (Il compositore di Musica)  
Cod. DTXQX01 - L. 42.000 - con istruzioni in italiano

Un ottimo programma sia per chi vuole avvicinarsi alla musica senza cognizioni di base, sia per chi deve studiare composizione direttamente sul programma e sia per chi utilizza il suono dal punto di vista tecnico ed ha interesse per tipo di frequenza, ecc.. Il programma prevede due opzioni: scrittura di partiture sul pentagramma: inserendo battuta per battuta è possibile scrivere partiture composte anche da tre voci e persino stamparne gli spartiti (in questo caso è necessaria una stampante e l'interfaccia RS232 o simile). L'altra opzione prevede la realizzazione delle melodie prendendo in esame gli Hertz di una singola nota.

**TI Invaders** (Invasori TI)  
Cod. DTXMX27 - L. 30.000

Il mondo viene attaccato da numerose creature odiose provenienti dallo spazio. Devi fare affidamento sulla tua agilità mentale e fisica per distruggere le creature multicolori con i missili a disposizione.

**Parsec**  
Cod. DTXMX21 - L. 42.000

PARSEC è la tua astronave, con la quale combatterai gli alieni ribelli e le loro navicelle, buttandole fuori rotta e emetterai un fuoco dal tuo laser che fa inaridire. Tenterai di sopravvivere alla "cintura" di asteroidi mortali. Per attivare le capacità sonore di questa cartuccia, puoi utilizzare lo Speech Synthesizer.

**Soccer** (Gioco del Calcio)  
Cod. DTXMX23 - L. 37.000

Inutile spendere troppe parole per questo apprezzatissimo programma. Cinque giocatori per squadra, con passaggi, tiri, intercettamenti, parate, cariche e molte altre tattiche del calcio. Potete persino avere il replay immediato di una rete, al rallentatore.

**Car Wars** (Guerre di Auto)  
Cod. DTXMX06 - L. 30.000

Se ti emozionano le corse automobilistiche, abbinare alla possibilità di battere l'avversario con astute manovre, ecco il programma che fa per te. Cimenta la tua velocità ed abilità contro il computer mentre cerchi di avanzare sul circuito, evitando incidenti che farebbe saltare il tuo bolide fuori pista.

E ora un'altra eccezionale proposta:

**Game Writer's Pack 1 & 2**  
Cod. DTXIX13 - L. 35.000  
(prezzo per le 2 confezioni)

Un'ottima realizzazione didattica divisa in due parti autonome ma complementari che insegna i principi (e li approfondisce) della programmazione BASIC, tramite la programmazione di videogiochi.

Molto ben strutturata e comprensibile, utilizza un linguaggio semplice ed elementare. In ogni confezione è compresa una cassetta con i giochi di cui si parla nel corso della dissertazione. In totale 6 giochi per confezione. La trattazione è basata sul TI-BASIC anche se non disdegna l'EXTENDED BASIC.

**Fun - Pac** (1-2-3)  
Cod. DTXMX36 - L. 27.000  
(prezzo per 3 cassette)

Tre cassette (non acquistabili separatamente), con 8 giochi complessivi ben strutturati ed avvincenti. Dalla corsa dei cavalli al gioco del golf. Dal salvataggio di scienziati dalla contaminazione nucleare alla navicella Supernova, un ottimo gioco "adventure"

**E INFINE...**

**Editor/Assembler**  
Cod. DTXAX01 - L. 180.000

L'eccezionale programma che ti permette di programmare in linguaggio di programmazione TSM 9900 è dà accesso diretto a tutte le caratteristiche del sistema tra cui audio, voce, grafica ed I/O (Entrata/Uscita), oltre a mettere a disposizione la massima rapidità grazie al microprocessore a 16 bit del computer. Le routine in linguaggio di programmazione possono essere svolte come programmi indipendenti oppure collegate a programmi TI BASIC o TI BASIC ESTESO con l'impiego di una chiamata di sottoroutine. Oltre al modulo SSS, il programma comprende due

floppy-disk (uno dei quali contiene il gioco TOMBSTONE CITY) e il Manuale Utente che dà tutte le informazioni sulla composizione del software. L'EDITOR/ASSEMBLER richiede i seguenti optional:

— **Floppy disk drive Box**  
Cod. ETXOX04 - L. 480.000

— **Peripheral Expansion Box**  
Cod. ETXOX01 - L. 320.000

— **32 K Expansion Card**  
Cod. ETXOX02 - L. 340.000

— **Drive Control Card**  
Cod. ETXOX03 - L. 440.000

**N. B.** È possibile acquistare insieme gli optional ETXOX01 - ETXOX02 - ETXOX03 - ETXOX04, al prezzo complessivo di L. 1.250.000 anziché L. 1.580.000. In questo caso, il codice da indicare per l'acquisto dei 4 pezzi è ETXOX05

e, infine, freschi di stampa, scontati del 10% sul prezzo di copertina:

— **Paper Book** - TI-99/4A (con cassetta) - editore J.soft  
Cod. ASOC007 - L. 16.200

— **35 Programmi per il TI-99/A4** - editore MUZZIO  
Cod. AMUCO24 - L. 9.000

Una selezione dei migliori programmi tratti dal nostro settimanale PAPER SOFT, scelti accuratamente sulla base della loro originalità e fantasia.

Programmi divertenti e didattici, dal labirinto al tennis, dagli anagrammi alle conversioni metriche e tanti altri ancora.

**RITAGLIA ED**

**INVIA**

**IL TAGLIANDO**

**A J.SOFT**

Spett. J.soft - Viale Restelli, 5 - 20124 Milano (MI)  
Tel. 02/6888228-683797-6880841/2/3/

Ordino il seguente software/libri per il TI-99/4A:

cod. .... cod. .... cod. ....  
cod. .... cod. .... cod. ....  
cod. .... cod. .... cod. ....

per un totale di L. .... + L. 2.500 per contributo fisso di spese di spedizione.

Scelgo la seguente modalità di pagamento:

- pagherò in contrassegno al postino
- allego assegno (o contanti)
- verso l'importo sul C.C.P. n. 19445204 intestato a J.soft (allego ricevuta)

Nome .....

Cognome .....

Via ..... n. ....

CAP ..... Città ..... Prov. ....

# Statistica a più dimensioni

Impariamo la statistica con l'aiuto  
del nostro Apple

di Roberto Brunialti

In tutte le situazioni in cui siano implicate più variabili, si può rendere necessaria la semplificazione del modello, consistente nella eliminazione di quelle che risultano ridondanti ai fini descrittivi del campione o di una variabile di particolare interesse per il ricercatore.

Generalmente si hanno due situazioni distinte:

**a)** quella in cui si sia interessati ad una o più variabili particolari, funzioni di una batteria di altre variabili indipendenti;

**b)** quella in cui non si privilegia alcuna delle variabili sulle altre, volendo sapere quali, fra tutte, contribuiscano maggiormente alla descrizione della popolazione da cui sono stati estratti i campioni, quali sono, cioè, le variabili più "variabili" delle altre.

Un esempio ci può essere fornito da un sondaggio statistico ove si eseguano misurazioni su insiemi di variabili scelte arbitrariamente al fine di vedere se nella popolazione saggiata, esistono tipologie particolari, suscettibili, mediante ulteriori analisi, di individuazione.

In questo caso non esistono variabili prioritarie, essendo nostro interesse individuare una struttura (se esiste) soggettiva, ma preventivamente ignorata, contrariamente al primo caso dove, invece si suppone esistere e si cerca conferma di una situazione.

Nel primo caso si procederà, quindi induttivamente, mentre nel secondo deduttivamente.

Pur rispondendo ad esigenze diverse, è interessante notare come il primo metodo si sia storicamente sviluppato prima dell'altro. Del resto la statistica nasce in casa delle scienze naturali e fisiche, in un secolo, quello scorso, in cui il positivismo supponeva tutto preesistente e disponibile all'indagine.

Al più il problema era trovare i mezzi fisici e logici per accedere alla conoscenza, cui attingere a piene mani.

È perciò comprensibile l'istintiva tendenza all'utilizzo di indagini "indirizzate ai dati", supporto alle teorie dello sperimentatore che li aveva rilevati.

Più recente è, invece, la disponibilità all'indagine "indirizzata al modello", in cui si può compiere una indagine preventiva, conoscitiva della realtà che interessa, con mente sgombra da eventuali presupposti riduttivi. Certamente questa tendenza è dovuta anche all'af-

fermarsi dell'uso del calcolatore, che rende possibili analisi, spesso ben supportate dalla teoria, ma necessitanti di un gravoso lavoro di calcolo.

Ma ritorniamo, dopo questa digressione, all'argomento oggetto del presente articolo.

Si faccia conto di aver compiuto misurazioni su un certo numero ( $q$ ) di variabili in  $n$  campioni.

Vogliamo ridurre le variabili a  $p$  ( $p < q$ ), in maniera che, nel caso di ulteriori rilevamenti, si possa ridurre il lavoro di prelevamento e di descrizione.

Se siamo nella condizione di cui al precedente punto a) e ci interessa sa-

pere, mettiamo il caso, la quantità di antiparassitario da usare in funzione della quantità di pioggia settimanale, della altezza della pianta, della quantità di concime data e della umidità media settimanale; volendo eliminare quante variabili sia possibile, senza peggiorare la capacità predittiva del modello, useremo il noto metodo della ricerca dei parametri della funzione di regressione lineare.

## Regressione multipla, parziale

Il metodo è noto, quindi saremo assai concisi nella descrizione del metodo,

**Tabella 1 - Dati ricavati dall'esempio 1.2 del libro di Sadocchi, riportato in bibliografia. Alle tre variabili ( $x_1$ ,  $x_2$ , e  $y$ ), bisogna ricordarsi di aggiungere una prima colonna tutta di 1, per ottenere il valore  $b_1$ , conformemente alla notazione dell'algebra matriciale.**

Variabile "dummy"	$x_1$	$x_2$	$y$
1	40	9	14
1	22	15	4
1	23	18	3
1	41	11	13
1	42	17	16
1	37	8	12
1	33	8	10
1	30	12	8
1	20	13	4
1	27	3	17
1	18	15	2
1	26	5	10
1	24	7	10
1	30	4	12
1	10	10	5
1	8	20	2
1	38	6	9
1	10	4	3
1	4	12	1
1	17	3	5

soffermandoci di più sui parametri, sul significato dei coefficienti trovati e sulla scelta delle variabili esplicative da mantenere o eliminare.

Come noto si cerca una funzione del tipo:

$$(1) \quad y = b_2 x_1 + \dots + b_q x_q + b_1$$

dove i parametri  $b_i$  indicano di quanto, per un incremento unitario della variabile dipendente  $i$ -esima, viene incrementato il valore della variabile di interesse  $y$ .

I parametri sono quindi valori angolari e rendono conto della eventuale dipendenza funzionale (e non causale) della variabile  $y$  dalle variabili esplicative  $X_i$ . Il calcolo dei coefficienti della (1) viene fatto sulla matrice  $\underline{A}$  contenente i dati rilevati (le righe, ricordiamo, sono i pro-

fili individuali dei campioni e le colonne rappresentano i caratteri o variabili rilevate):

$$(2) \quad \begin{matrix} b_1 + b_2 x_{11} + \dots + b_q x_{1q} = y_1 \\ \vdots \\ b_1 + b_2 x_{n1} + \dots + b_q x_{nq} = y_n \end{matrix}$$

$$\Leftrightarrow \underline{A} b = y \Leftrightarrow b = \underline{A}^{-1} y$$

e, poiché  $\underline{A}$  è una matrice generalmente non quadrata, l'unica inversa che si potrà usare è quella di Moore-Penrose, valida per matrici rettangolari e calcolata come  $(\underline{A}' \underline{A})^{-1} \underline{A}'$  ( $-1$  è il simbolo della inversa non singolare ordinaria) da cui, con ovvio passaggio:

$$(3) \quad b = (\underline{A}' \underline{A})^{-1} \underline{A}' y$$

I valori trovati sono, naturalmente, solo

una stima campionaria, che può essere più correttamente trattata considerando anche i limiti fiduciali (gli intervalli entro i quali si ha una probabilità  $1-\alpha$  per  $\alpha$  arbitrario, di trovare il valore reale), calcolati come:

$$(4) \quad \overleftrightarrow{b}_i = b_{i \pm} z_{\alpha} S_b$$

con, per esempio,  $z_{0,05} = 1,96$  ed  $S$  scarto quadratico medio calcolabile come:

$$(5) \quad S_b^2 = (e'e / (n-q-1)) (\underline{A}' \underline{A})^{-1} e = (y - \underline{A}b)$$

Per verificare che ogni singolo coefficiente calcolato sia *significativamente* diverso da 0 con probabilità  $\alpha$  basterà accettare l'ipotesi (di significativa diversità) se  $b / \text{SQR}(S_b) \leq |t/\alpha/2|$  con  $t$  variabile distribuita secondo la  $t$ -Student con  $(n-q-1)$  gradi di libertà.

I coefficienti trovati però, pur rendendo conto da soli del *contributo netto* alla variabile di interesse (cioè eliminata la parte di correlazione esistente tra la stessa e le  $q-1$  variabili esplicative) sono distorti dall'unità di misura delle singole variabili  $x_i$ . Bisognerà, perciò, utilizzare i cosiddetti coefficienti  $\beta$ , facilmente derivabili dai coefficienti  $b$  della (3) mediante la seguente trasformazione:

$$(6) \quad \beta_i = b_i (S_i / S_y) \\ S = \text{SQR}(\text{varianza})$$

ottenendo dei valori confrontabili per la scelta delle variabili più significative che

t	Immatricolazioni x 1.000
1	100.7
2	107.1
3	95.5
4	115.3
5	100.5
6	99.0
7	115.3
8	113.6
9	Dato da "prevedere"
10	Dato da "ripetere"
...	.....

**Tabella 2 - Dati sulle immatricolazioni di automobili nei due anni, esempio numerico per l'autoregressione dei risultati. Il valore t è semplicemente un indice progressivo, dal 1° dato all'ultimo. Serve a "pesare" diversamente i dati, dando un maggior valore ai più recenti.**

#### Listato 1 - Il programma Multireg.

```

1 REM *****
2 REM *      MULTIREG      *
4 REM * ----- *
5 REM * BY R. BRUNIALTI *
6 REM *   ON 10.3.84   *
7 REM *****
8 REM
9 REM
10 REM
1000 REM *****
1010 REM MENU' PRINCIPALE
1020 REM *****
1030 HOME
1040 D$ = CHR$ (4)
1050 PRINT : PRINT
1060 FOR I = 1 TO 40:LN$ = LN$ +
    "-" : NEXT
1070 PRINT LN$

```

```

1080 PRINT : PRINT
1090 PRINT "      1 -- REGRESSIONE
MULTIPLA"
1100 PRINT
1110 PRINT "      2 - AUTOREGRESSI
ONE"
1120 PRINT : PRINT
1130 PRINT LN$
1140 INPUT "QUALE? ";N
1150 ON N GOSUB 10000,20000
1160 REM
2000 REM *****
2010 REM CALCOLO MARGINALI
2020 REM *****
2030 REM
2040 PM = R1 + 1:PV = R1 + 2
2050 FOR I = 1 TO C1
2060 FOR J = 1 TO R1
2070 A(PM, I) = A(PM, I) + A(J, I)
2080 A(PV, I) = A(PV, I) + A(J, I) *

```

## Seguito Ilistato Multireg

```

      A(J, I)
2090 NEXT
2100 NEXT
2110 FOR I = 1 TO R1
2120 A(PM, 0) = A(PM, 0) + Y(I)
2130 A(PV, 0) = A(PV, 0) + Y(I) * Y
      (I)
2140 Y(PM) = Y(PM) + Y(I)
2150 Y(PV) = Y(PV) + Y(I) * Y(I)
2160 NEXT
2170 Y(PV) = (Y(PV) - Y(PM) ^ 2 /
      R1)
2180 Y(PM) = Y(PM) / R1
2190 REM
3000 REM *****
3010 REM CALCOLO (A' * A)
3020 REM *****
3030 REM
3040 FOR I = 1 TO C1
3050 FOR J = 1 TO C1
3060 FOR K = 1 TO R1
3070 INV(I, J) = INV(I, J) + A(K, I)
      * A(K, J)
3080 NEXT : NEXT : NEXT
3090 REM
4000 REM *****
4010 REM CALCOLO (A' * A)^-1
4020 REM *****
4030 REM
4040 FOR K = 1 TO C1
4050 FOR J = 1 TO C1
4060 IF J = K THEN 4120
4070 INV(K, J) = INV(K, J) / INV(K,
      K)
4080 FOR I = 1 TO C1
4090 IF I = K THEN 4110
4100 INV(I, J) = INV(I, J) - INV(I,
      K) * INV(K, J)
4110 NEXT I
4120 NEXT J
4130 FOR I = 1 TO C1
4140 IF I = K THEN 4160
4150 INV(I, K) = - INV(I, K) / INV
      (K, K)
4160 NEXT I
4170 INV(K, K) = 1 / INV(K, K)
4180 NEXT K
5000 REM *****
5010 REM PRODOTTO
5020 REM *****
5030 REM
5040 FOR I = 1 TO C1
5050 FOR J = 1 TO R1
5060 A(I, 0) = A(I, 0) + A(J, I) * Y
      (J)
5070 NEXT : NEXT
5080 REM
6000 REM *****
6010 REM COEFF B
6020 REM *****
6030 REM
6040 FOR I = 1 TO C1
6050 FOR J = 1 TO C1
6060 A(0, I) = A(0, I) + INV(I, J) *
      A(J, 0)
6070 NEXT : NEXT
6080 REM
7000 REM *****
7010 REM ERR AB-Y
7020 REM *****
7030 REM
7040 REM
7050 REM *** VALORI TEORICI ***
7060 REM
7070 FOR I = 1 TO R1
7080 A(I, 0) = 0
7090 FOR J = 1 TO C1
7100 A(I, 0) = A(I, 0) + A(0, J) * A
      (I, J)
7110 NEXT : NEXT
7120 REM
7130 REM *** DIFFERENZA ***
7140 REM
7150 FOR I = 1 TO R1
7160 A(I, C) = Y(I) - A(I, 0)
7170 A(0, C) = A(0, C) + A(I, C)
7180 A(0, 0) = A(0, 0) + A(I, C) * A
      (I, C)
7190 NEXT
7200 A(PV, C) = (A(0, 0) - (A(0, C) ^
      2 / R1)) / (R1 - 1)
7210 A(PM, C) = A(0, C) / R1
7220 REM
7230 REM *** MEDIE, VAR, T-STAT *
      **
7240 REM
7250 SIG = A(0, 0) / (R1 - C1)
7260 FOR I = 1 TO C1
7270 BV(I) = SIG * INV(I, I)
7280 TV(I) = A(0, I) / SQR (BV(I)
      )
7290 A(PV, I) = (A(PV, I) - (A(PM, I)
      ) ^ 2 / R1)) / (R1 - 1)
7300 A(PM, I) = A(PM, I) / R1
7310 VT = VT + A(PV, I)
7320 NEXT
7330 DT = 1 - (A(0, 0) / Y(PV))
7340 Y(PV) = Y(PV) / (R1 - 1)
7350 REM
8000 REM *****

```

**Seguito listato Multireg**

```

8010 REM STAMPA RISULTATI
8020 REM *****
8030 REM
8040 A(PV,1) = 0
8050 PRINT "ORDER";: PRINT TAB(
14)"MEDIA";: PRINT TAB( 30)
"VAR": PRINT
8060 PRINT " Y";: PRINT TAB( 1
1)Y(PM);: PRINT TAB( 27)Y(P
V)
8070 FOR I = 1 TO C1
8080 PRINT " ";I;
8090 PRINT TAB( 11)A(PM, I);: PRINT
TAB( 27)A(PV, I)
8100 NEXT
8110 PRINT
8120 PRINT TAB( 14)"COEFF";: PRINT
TAB( 30)"COEFF"
8130 PRINT TAB( 14)" B ";: PRINT
TAB( 30)"BETA"
8140 PRINT
8150 FOR I = 1 TO C1
8160 PRINT " ";I;
8170 PRINT TAB( 11)A(O, I);: PRINT
TAB( 27)A(O, I) * A(PV, I) /
Y(PV)
8180 NEXT
8190 PRINT
8200 PRINT TAB( 14)"STAT";: PRINT
TAB( 30)"VAR"
8210 PRINT TAB( 14)" T";: PRINT
TAB( 30)" B"
8220 PRINT
8230 FOR I = 1 TO C1
8240 PRINT " ";I;
8250 PRINT TAB( 11)TV(I);: PRINT
TAB( 27)BV(I)
8260 NEXT
8270 PRINT
8280 PRINT "TOT VAR X= ";VT
8290 PRINT "COEFF DET= ";DT
8300 PRINT
8310 PRINT "(E*E)/(P-Q) =
";SIG
8320 PRINT "MEDIA DEGLI ERRORI=
";A(PM,C)
8330 PRINT "VAR. DEGLI ERRORI=
";A(PV,C)
8340 END
10000 REM *****
10010 REM MULTIREGRESSIONE
10020 REM *****
10030 HOME :PT = 1
10040 PRINT LN$
10050 PRINT : PRINT
10060 PRINT "NOME FILE DATI VARI

```

```

ABILI INDIPENDENTI"
10070 PRINT
10080 INPUT " ";A$
10090 PRINT
10100 PRINT "NOME FILE DATI VARI
ABILE DIPENDENTE"
10110 PRINT

```

**Figura 1 - Esempio di come saranno disposti i dati alla fine dell'elaborazione; eventuali programmi che necessitano dei dati suddetti possono essere "accodati" alla fine del programma Multireg tenendo conto della figura.**

$$y'_i = \sum_{j=1}^q x_{ij} b_j$$

$$e = y - y^*$$

$\bar{x}_i$  = media var. i-esima

$S^2_i$  = varianza var. i-esima

Numero righe	$\sum_{i=1}^n e_i^2$	$b_1$	$b_2$	$b_3$	$\sum_{i=1}^n e_i^2$
0					
1	$y_1^*$	1	40	9	$e_1$
2	$y_2^*$	1	22	15	$e_2$
3	$y_3^*$	1	23	18	$e_3$
4	$y_4^*$	1	41	11	$e_4$
5	$y_5^*$	1	42	17	$e_5$
6	$y_6^*$	1	37	8	$e_6$
7	$y_7^*$	1	33	8	$e_7$
8	$y_8^*$	1	30	12	$e_8$
9	$y_9^*$	1	20	13	$e_9$
10	$y_{10}^*$	1	27	3	$e_{10}$
11	$y_{11}^*$	1	18	15	$e_{11}$
12	$y_{12}^*$	1	26	5	$e_{12}$
13	$y_{13}^*$	1	24	7	$e_{13}$
14	$y_{14}^*$	1	30	4	$e_{14}$
15	$y_{15}^*$	1	10	10	$e_{15}$
16	$y_{16}^*$	1	8	20	$e_{16}$
17	$y_{17}^*$	1	38	6	$e_{17}$
18	$y_{18}^*$	1	10	4	$e_{18}$
19	$y_{19}^*$	1	4	12	$e_{19}$
20	$y_{20}^*$	1	17	3	$e_{20}$
		—	—	—	—
	$y^*$	$x_1$	$x_2$	$x_3$	$e$
	$S^2_{y^*}$	$S^2_1$	$S^2_2$	$S^2_3$	$S^2_e$
	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
		<b>Numero colonne</b>			

verrà effettuata scartando quelle che:  
**1)** hanno i coefficienti  $b_i$  non significativi diversi da 0;

**2)** hanno i coefficienti  $\beta_i$  più bassi di un certo livello prefissato.

Può essere anche di un certo interesse calcolare il coefficiente di determinazione del modello,  $R^2$  che denota la percentuale della varianza della variabile  $y$  spiegata dall'insieme delle variabili  $x_i$ . Tanto più elevato il valore, (compreso tra 0 ed 1) tanto più buona è l'approssimazione del modello a quello deterministico (in cui, cioè, il valore di  $y$  è completamente fissato, noti quelli delle  $x_i$ ), mai occorrente nella pratica, corrispondente a  $R^2 = 1$ .

### Autoregressione ed analisi delle serie storiche

Una applicazione che usa i risultati raggiunti dalla regressione lineare multipla a scopi predeittivi, assai interes-

sante per la semplicità e la efficacia dei risultati, è quella nota come "autoregressione".

Qualora i dati vengano campionati in funzione del tempo, ad intervalli costanti, (o sia comunque possibile ricondurli sotto tale forma) possiamo considerare gli stessi come realizzazioni di una variabile dipendente dai propri valori pregressi che si comportano come variabili esplicative vere e proprie.

Per esempio, se abbiamo dei dati riguardanti il numero di immatricolazioni di macchine lungo un certo periodo (1977-78), rilevato a scadenza trimestrale, possiamo costruire un modello di regressione mediante il quale prevedere l'ammontare delle nuove immatricolazioni in un prossimo intervallo di tempo futuro.

In tabella 2 vengono riportati i valori tratti da un esempio comparso nell'articolo di J. Hudson riportato in bibliografia.

Il problema che vogliamo risolvere è il seguente:

*possiamo prevedere il numero di nuove immatricolazioni trimestrali lungo tutto l'arco dell'anno 1979?*

La risposta è affermativa. Si suppone infatti che i dati contengano una certa quantità di informazione sul "trend" generale che essi descrivono. Anche la logica ci porta a considerare non peregrina l'idea che le immatricolazioni in un dato tempo  $t$  dipendano dal numero delle stesse, negli intervalli di tempo immediatamente precedenti.

La funzione ricercata sarà del tipo:

$$(7) \quad y_t = b_1 + tb_2 + y_{t-1}b_3 + \dots + y_{t-q}$$

dove  $q$  è il numero degli intervalli pregressi che intendiamo usare nella determinazione della variabile di interesse.

La scelta del valore di  $q$  dipende, essenzialmente dal tipo di dati a cui si applica l'analisi. Non conviene sceglierlo troppo grande se siamo interessati ai

#### Seguito listato Multireg

```

10120 INPUT " ";Y$
10130 PRINT : PRINT
10140 PRINT LN$
10150 PRINT D$;"OPEN";A$
10160 PRINT D$;"READ";A$
10170 INPUT ND: INPUT C1: INPUT
      R1
10180 PRINT D$;"OPEN";Y$
10190 PRINT D$;"READ";Y$
10200 INPUT ND: INPUT C2: INPUT
      R2
10210 IF C2 > 1 THEN GOSUB 5004
      0
10220 IF R1 < > R2 THEN GOSUB
      50050
10230 PRINT D$;"READ";A$
10240 FOR I = 1 TO R1
10250 INPUT TP: IF TP < > 1 THEN
      EF = 1: GOTO 10270
10260 NEXT
10270 GOSUB 50200
10280 C = C1 + 1
10290 DIM A(R1 + 2,C),Y(R1 + 2),
      INV(C1,C1),BV(C1),TV(C1)
10300 IF EF THEN FOR I = 1 TO R
      1:A(I,1) = 1: NEXT
10310 FOR I = PT TO C1
10320 FOR J = 1 TO R1
10330 INPUT A(J,I)

```

```

10340 NEXT : NEXT
10350 PRINT D$;"READ";Y$
10360 FOR I = 1 TO R1
10370 INPUT Y(I)
10380 NEXT
10390 PRINT D$;"CLOSE"
10400 RETURN
10410 REM
20000 REM *****
20010 REM AUTOREGRESSIONE
20020 REM *****
20030 HOME
20040 PRINT LN$;
20050 INPUT "NOME FILE DATI: ";A
      $
20060 PRINT LN$;
20070 PRINT : PRINT
20080 INPUT "NUMERO DEI PERIODI
      PREGRESSI:";C1
20090 PRINT : PRINT : PRINT LN$
20100 PRINT D$;"OPEN";A$
20110 PRINT D$;"READ";A$
20120 INPUT ND: INPUT NC: INPUT
      NR
20130 R1 = NR - C1:C1 = C1 + 2
20140 IF NOT (R1 > C1) THEN GOSUB
      50060
20150 IF NC < > 1 THEN GOSUB 5
      0040
20160 DIM BV(C1),TV(C1),A(R1 + 2
      ,C1 + 1),Y(R1 + 2),INV(C1,C1)

```

valori immediatamente successivi ai dati più recenti in nostro possesso; Nell'esempio in questione, essendo il risultato un valore medio di tendenza, se siamo interessati al valore trimestrale converrà scegliere  $q = 4$ ; se siamo invece interessati ai valori annuali, converrà scegliere valori maggiori. I valori ricavabili dalla (7) all'aumentare dell'indice  $t$  tendono a degenerare, fornendo dati poco attendibili. Ciò è comprensibile qualora si consideri che l'informazione contenuta nei dati, renda chiara una tendenza in un *intervallo prossimo* di tempo.

Per quanto detto dalla (7), si deriva che la nostra matrice sarà quella deducibile dal sistema di equazioni:

$$(8) \begin{matrix} b_1 + tb_2 + y_{t-1}b_3 + \dots + y_{t-q}b_{q+2} = y_t \\ \vdots \\ b_1 + (q+1)b_2 + y_qb_3 + \dots + y_1b_{q+2} = y_{q+1} \end{matrix}$$

$t$  = numero dei dati

$$n = t - q$$

immediatamente trasformabile nella (2) (vedere, per maggior chiarimento la figura 2).

Si ricorda che, poiché il numero delle equazioni ( $n$ ) è pari a  $t - q$  e dato che bisogna determinare i parametri di  $q + 2$  variabili, comprendono i coefficienti  $b_i$ , e  $b_2$  deve essere  $t > q + 2$ . In caso contrario non si potrebbero calcolare tutti i valori per l'insufficiente numero di equazioni nel sistema di figura 4.

### Il programma Multireg

Quanto detto sino ad ora è facilmente applicabile nella pratica mediante il programma Multireg, di uso assai semplice. Esso permette il calcolo dei parametri delle funzioni di regressione multilineari, insieme a tutti gli altri valori di interesse statistico.

Per selezionare una delle due metodologie, basta inserire un 1 o un 2 alla richiesta formulata dal menu principale. Se si sceglie l'opzione 1 (multiregressione) bisognerà specificare, su richiesta dello stesso programma:

**1)** nome del file in cui giacciono i dati delle variabili *indipendenti*, sotto forma di matrice, scritta nel formato di cui alla figura 3, mediante il programma Edit;

**2)** nome del file in cui giacciono i dati della variabile *dipendente*, sotto forma di matrice  $n \times 1$ .

Fatto quanto richiesto, dopo un'attesa, variabile a seconda della quantità dei dati raccolti, si avrà un output completo. Consigliamo, qualora si disponga di una stampante di inserire come prima linea della routine "Stampa dei risultati" una linea come quella che segue:

```
xxxx PRINT D$;"PR#";NS
```

inoltre la variabile NS (riga 10) deve essere settata con il numero dello slot in

#### Seguito Ilistato Multireg

```
)
20165 FOR I = 3 TO C1
20166 INPUT A(1, I)
20167 NEXT
20170 FOR I = 1 TO R1
20180 A(I, 1) = 1
20190 A(I, 2) = I + 2
20200 INPUT Y(I)
20210 NEXT
20220 FOR I = 2 TO R1
20230 A(I, C1) = Y(I - 1)
20240 FOR J = C1 - 1 TO 3 STEP -
1
20250 A(I, J) = A(I - 1, J + 1)
20260 NEXT : NEXT
20360 RETURN
50000 REM *****
50010 REM ERROR MSG
50020 REM *****
50030 REM
50040 PRINT "VETTORE ";Y$;" HA P
IU' DI 1 COLONNA": END
50050 PRINT "RIGHE MAT ";A$;" <
> RIGHE VET ";Y$: END
50060 PRINT "I PERIODI PREGRESSI
SONO TROPPI PER I DATI IN
POSSESSO": END
50200 IF NOT EF THEN GOTO 5026
0
```

```
50210 PRINT "*** WARNING ***": PRINT
50220 PRINT "NELLA COLONNA 1 DEL
LA MAT ";A$
50230 PRINT "NON E' PRESENTE LA
VARIABILE DUMMY '1'"
50240 IF I < > 1 THEN PRINT "N
ON E' POSSIBILE RIMEDIARE": END
50250 PRINT "TENTO DI INTERVENIR
E INSERENDO UNA": PRINT "COL
ONNA DI '1' IN PRIMA POSIZIO
NE"
50260 PRINT D$;"CLOSE";A$
50270 PRINT D$;"OPEN";A$
50280 PRINT D$;"READ";A$
50290 INPUT ND: INPUT C1: INPUT
R1
50300 IF NOT EF THEN RETURN
50310 C1 = C1 + 1:PT = 2
50320 RETURN
```

## Listato 2 - Il programma PCA.

```

1 REM *****
2 REM *      PCA PROGRAM      *
3 REM *      -----      *
4 REM *      BY R. BRUNIALTI  *
5 REM *      ON 10.5.84      *
6 REM *****
7 REM
8 REM
9 REM
10 FOR I = 1 TO 39:LN$ = LN$ + "
    -": NEXT
20 HOME
30 D$ = CHR$(4)
40 REM
1000 REM *****
1010 REM      MENU'
1020 REM *****
1030 REM
1040 PRINT LN$
1050 INPUT "FILE DATI INPUT : ";
    N$
1060 PRINT LN$
1070 PRINT
1080 PRINT "FILES DI OUTPUT SU D
    ISCHETTO (S/N)"
1090 PRINT
1100 PRINT N$;".R";
1110 HTAB (20): GET FR$
1120 IF FR$( ) "S" AND FR$( )
    "N" THEN GOTO 1110
1130 PRINT FR$
1140 PRINT N$;".AVL";
1150 HTAB 20: GET FL$
1160 IF FL$( ) "S" AND FL$( )
    "N" THEN GOTO 1150
1170 PRINT FL$
1180 PRINT N$;".AVT";
1190 HTAB 20: GET FT$
1200 IF FT$( ) "S" AND FT$( )
    "N" THEN GOTO 1190
1210 PRINT FT$
1220 PRINT N$;".S";
1230 HTAB 20: GET FS$
1240 IF FS$( ) "S" AND FS$( )
    "N" THEN GOTO 1230
1250 PRINT FS$
1260 PRINT LN$: PRINT LN$
1270 GOSUB 10000
1280 DIM VL(NC), VT(NC, NC), R(NC, N
    C), S(NC, NC)
1290 REM
2000 REM *****
2010 REM      CALCOLA R
2020 REM *****
2030 REM
2040 FOR I = 1 TO NC
2050 FOR J = 1 TO NR
2060 A(O, I) = A(O, I) + A(J, I)
2070 A(I, O) = A(I, O) + A(J, I) * A
    (J, I)
2080 NEXT
2090 REM *** VARIANZE ***
2100 A(I, O) = SQR ((A(I, O) - A(O
    , I) * A(O, I) / NR) / (NR - 1
    ))
2110 REM *** MEDIE ***
2120 A(O, I) = A(O, I) / NR
2130 NEXT
2140 REM *** STANDARDIZ.***
2150 FOR I = 1 TO NR
2160 FOR J = 1 TO NC
2170 A(I, J) = (A(I, J) - A(O, J)) /
    A(J, O)
2180 NEXT : NEXT
2190 REM *** PRODOTTO ***
2200 FOR J = 1 TO NC
2210 FOR I = J TO NC
2220 FOR K = 1 TO NR
2225 R(J, I) = R(J, I) + A(K, J) * A
    (K, I)
2230 NEXT
2240 R(J, I) = R(J, I) / (NR - 1):R
    (I, J) = R(J, I)
2245 NEXT : NEXT
2250 IF FR$ = "S" THEN GOSUB 20
    040
3000 REM *****
3005 REM      CALCOLA VT&VL
3010 REM *****
3015 REM
3020 REM *** INIZIALIZZA ***
3025 REM
3030 S1 = 8
3035 FOR I = 1 TO NC
3040 VL(I) = 1
3045 VT(I, I) = 1
3050 NEXT I
3055 Z = 2 * S1
3060 T1 = 1 / (10 ^ Z)
3065 R = 5 * NC ^ 2
3070 R1 = 0
3075 T2 = .1
3080 N1 = NC - 1
3085 REM *** ROTAZIONE **
3090 X1 = 0
3095 FOR K = 1 TO N1
3100 K1 = K + 1
3105 FOR L = K1 TO NC
3110 A1 = R(K, K)
3115 A2 = R(K, L)
3120 A3 = R(L, L)
3125 X = A2 * A2 / (A1 * A3)

```

Seguito Ilistato PCA

```

3130 IF X > X1 THEN 3140
3135 GOTO 3145
3140 X1 = X
3145 IF X < T2 THEN 3360
3150 R1 = R1 + 1
3155 IF A1 = A3 THEN 3180
3160 Z = .5 * (A1 - A3) / A2
3165 Z1 = 1 + 1 / (Z * Z)
3170 T = - Z * (1 + SQR (Z1))
3175 GOTO 3185
3180 T = 1
3185 C = 1 / SQR (1 + T * T)
3190 S = C * T
3195 S2 = S * S
3200 C2 = C * C
3205 R(K, L) = 0
3210 REM
3215 A0 = 2 * A2 * C * S
3220 R(K, K) = A1 * C2 + A0 + A3 * S2
3225 R(L, L) = A1 * S2 - A0 + A3 * C2
3230 REM
3235 FOR I = 1 TO NC
3240 IF I < K THEN 3255
3245 IF I > K THEN 3275
3250 GOTO 3325
3255 A0 = R(I, K)
3260 R(I, K) = C * A0 + S * R(I, L)

```

```

3265 R(I, L) = - S * A0 + C * R(I, L)
3270 GOTO 3325
3275 IF I < L THEN 3290
3280 IF I > L THEN 3310
3285 GOTO 3325
3290 A0 = R(K, I)
3295 R(K, I) = C * A0 + S * R(I, L)

3300 R(I, L) = - S * A0 + C * R(I, L)
3305 GOTO 3325
3310 A0 = R(K, I)
3315 R(K, I) = C * A0 + S * R(L, I)

3320 R(L, I) = - S * A0 + C * R(L, I)
3325 NEXT I
3330 REM
3335 FOR I = 1 TO NC
3340 U0 = VT(I, K)
3345 VT(I, K) = C * U0 + S * VT(I, L)
3350 VT(I, L) = - S * U0 + C * VT(I, L)
3355 NEXT I
3360 NEXT L
3365 NEXT K
3370 IF X1 < T1 THEN 3390
3375 IF R1 > R THEN GOSUB 50040

```

```

3380 T2 = .1 * X1
3385 GOTO 3090
3390 REM *** NORMALIZZA ***
3395 FOR I = 1 TO NC
3400 FOR J = 1 TO NC
3405 VT(I, J) = VT(I, J) * VL(I)
3410 NEXT J
3415 NEXT I
3420 FOR I = 1 TO NC
3425 VL(I) = R(I, I)
3430 NEXT I
3435 REM
3440 FOR I = 1 TO N1
3445 I1 = I + 1
3450 Z = VL(I)
3455 M = I
3460 FOR J = I1 TO NC
3465 IF Z < VL(J) THEN 3480
3470 Z = VL(J)
3475 M = J
3480 NEXT J
3485 VL(M) = VL(I)
3490 VL(I) = Z
3495 FOR J = 1 TO NC
3500 Z = VT(J, I)
3505 VT(J, I) = VT(J, M)

```

Figura 2 - Matrice A, b e y ricavata dalla tabella dati n.2, per maggiori chiarimenti vedere formule (7) ed (8).

1	8	115.3	99.0	.	b <sub>1</sub>	113.6
1	7	99.0	100.5		b <sub>2</sub>	115.3
1	6	100.5	115.3		b <sub>3</sub>	99.0
1	5	115.3	95.5		b <sub>4</sub>	100.5
1	4	95.5	107.1		b <sub>5</sub>	115.3

Figura 3 - Formato di immagazzinamento delle matrici su dischetto. Vanno ordinate per colonne, facendo precedere ai dati una label con le seguenti informazioni: ND = sempre uguale a 2 (numero delle dimensioni) - NC = numero delle colonne - NR = numero delle righe.

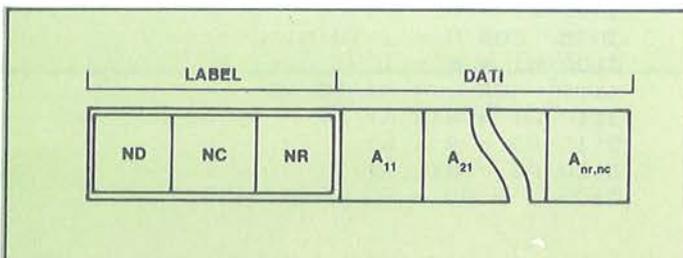
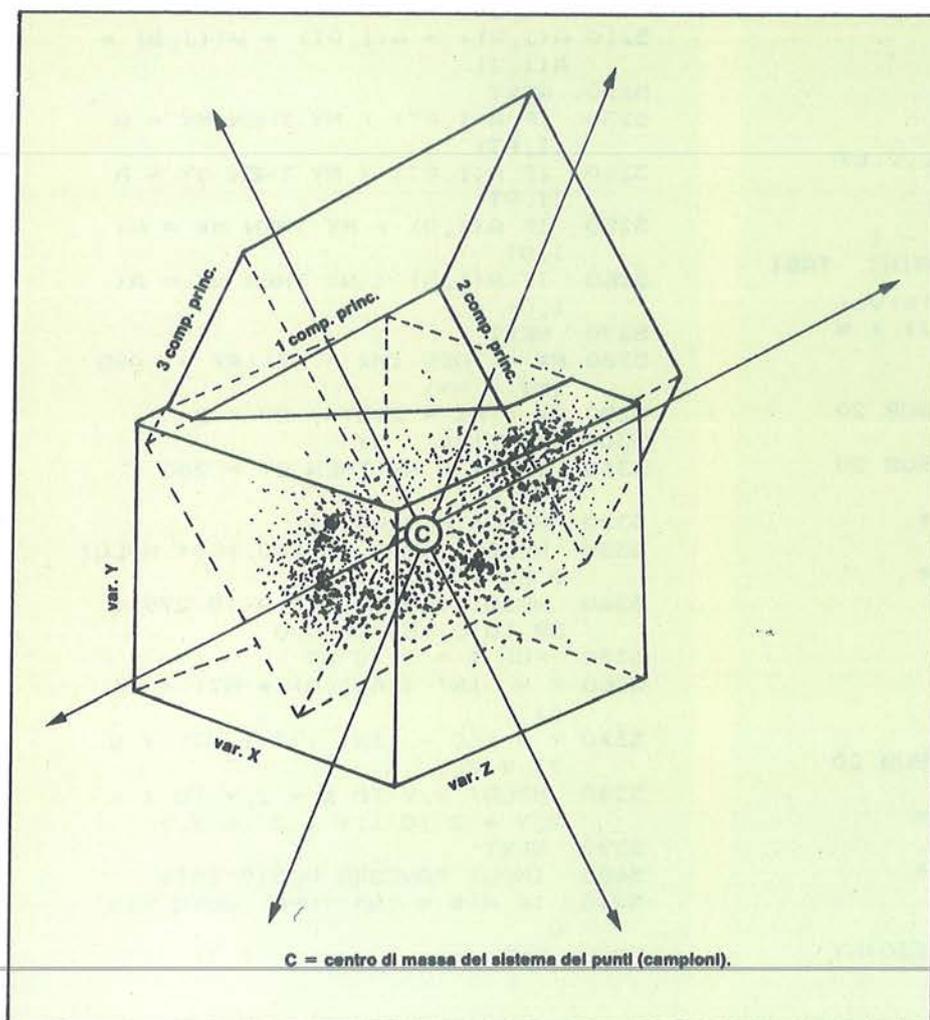


Figura 4 - Esempificazione geometrica della PCA (caso  $q=3$ ) per le variabili  $x, y$  e  $z$ . Il nuovo sistema di riferimento sarà dato dalle tre componenti (o assi) principali, orientati conformemente agli assi passanti per il centro di massa,  $C$ , del sistema di punti-campione.



cui è inserita l'interfaccia, agendo anche da flag per la stampa di ulteriori dati, non necessari nella versione con uscita su video.

Qualora si scelga l'opzione 2 (autoregressione) bisognerà specificare:

**1)** nome del file contenente i dati della variabile  $Y$  in oggetto (dimensioni  $t \times 1$ , sempre nel formato come in figura 5;

**2)** numero ( $q$ ) dei periodi pregressi che vengono considerati nella espressione (7).

Anche in questo caso si avranno gli output forniti nel caso precedente.

### Suggerimenti per l'adattamento ad altri computer del programma Multireg

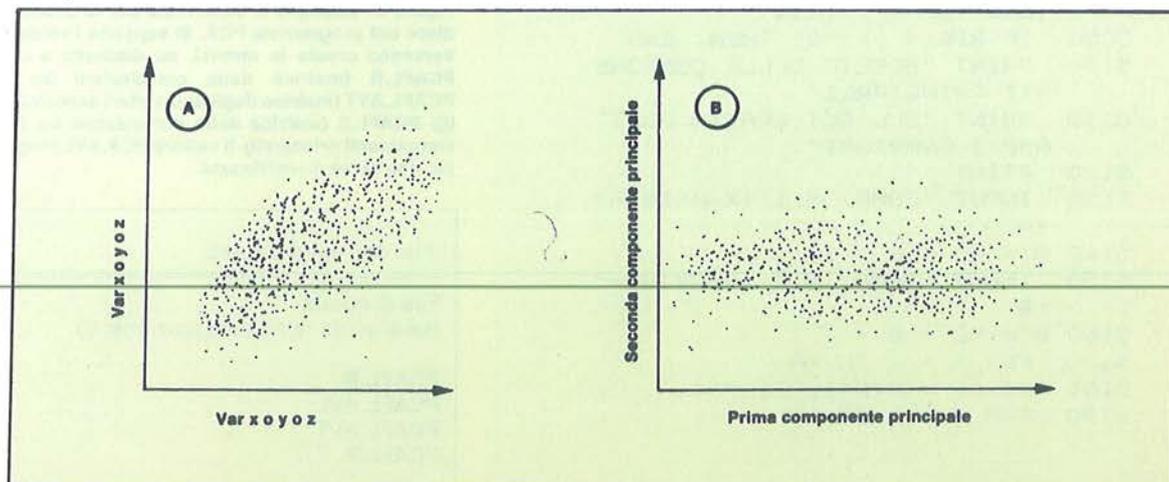
Il programma è stato scritto utilizzando le istruzioni generiche del BASIC Microsoft, tranne che per le routine 10000 e 20000, dove si utilizzano comandi del DOS 3.3.

Qualora si volesse utilizzare il programma su altri computer, bisogna che a partire dalla linea 1000:

**1)** i dati siano inseriti nelle matrici e vettori, dimensionati nel modo seguente:

- BV (C1) vettore delle varianze dei coefficienti  $b_i$ ;
- TV (C1) vettore dei valori  $t$  per il test di significanza dei  $b_i$ ;
- A (R1 + 2, C1 + 1) matrice dei dati, deve avere già i dati inseriti;
- Y (R1) vettore variabili dipendenti, deve avere i dati inseriti;
- INV (C1, C1) matrice in cui verrà calco-

Figura 5 - Considerando la figura 4, in A appaiono le proiezioni dei punti-campione, scegliendo coppie di assi costituiti dalle variabili originarie. In B il risultato dopo la trasformazione degli stessi con la PCA. Se si considera che le variabili sono solo tre si può apprezzare l'incremento del potere di distinzione dei clusters.



Seguito listato PCA

```

3510 VT(J,M) = Z
3515 NEXT J
3520 NEXT I
3525 I = 0
3530 PRINT
3535 PRINT "N.....L.....L/
Q.....CUM....."
3540 PRINT
3545 FOR J = NC TO 1 STEP - 1
3550 I = I + 1: PRINT I;: PRINT TAB(
3)VL(J);: PRINT TAB( 16)VL(
J) / NC;:AC = AC + VL(J) / N
C: PRINT TAB( 30)AC
3555 NEXT
3560 IF FT$ = "S" THEN GOSUB 20
130
3565 IF FL$ = "S" THEN GOSUB 20
220
4000 REM *****
4010 REM CALCOLA S
4020 REM *****
4030 REM
4040 FOR I = 1 TO NC
4050 TP = SQR (VL(I))
4060 FOR J = 1 TO NC
4070 S(J,I) = TP * VT(J,I)
4080 NEXT : NEXT
4090 IF FS$ = "S" THEN GOSUB 20
300
5000 REM *****
5010 REM PLOT
5020 REM *****
5030 REM
5040 PT = NC + 1
5050 MX = - 1E30:MY = - 1E30:NY
= 1E30:NX = 1E30
5060 PRINT LN$
5070 PRINT
5080 INPUT "VUOI PLOTTARE I CAMP
IONI (S/N) ?";RI$
5090 IF RI$ < > "S" THEN END
5100 PRINT "SCELTA DELLE COMPONE
NTI PRINCIPALI"
5110 PRINT "SUL CUI SPAZIO PLOTT
ARE I CAMPIONI"
5120 PRINT
5130 INPUT "COMP. N.1 (X-AXIS)="
;A
5140 A = NC - A + 1
5150 INPUT "COMP. N.2 (Y-AXIS)="
;B
5160 B = NC - B + 1
5170 FOR I = 1 TO NR
5180 A(I,0) = 0:A(I,PT) = 0
5190 FOR J = 1 TO NC

```

```

5200 A(I,0) = A(I,0) + VT(J,A) *
A(I,J)
5210 A(I,PT) = A(I,PT) + VT(J,B) *
A(I,J)
5220 NEXT
5230 IF A(I,PT) > MY THEN MY = A
(I,PT)
5240 IF A(I,PT) < NY THEN NY = A
(I,PT)
5250 IF A(I,0) > MX THEN MX = A(
I,0)
5260 IF A(I,0) < NX THEN NX = A(
I,0)
5270 NEXT
5280 RX = ABS (MX - NX):RY = ABS
(MY - NY)
5290 RX = RX * 2:RY = RY * 2
5300 ST = 160 / RY
5310 IF RX > RY THEN ST = 280 /
RX
5320 HGR : HCOLOR= 3
5330 HPLOT 140,0 TO 140,160: HPLOT
0,80 TO 279,80
5340 HPLOT 0,0 TO 279,0 TO 279,1
59 TO 0,159 TO 0,0
5350 FOR I = 1 TO NR
5360 X = INT ((A(I,0) * ST) + 14
0)
5370 Y = 160 - INT ((A(I,PT) * S
T) + 80)
5380 HPLOT X,Y TO X + 2,Y TO X +
2,Y + 2 TO X,Y + 2 TO X,Y
5390 NEXT
5400 INPUT "ANCORA PLOT?";RI$
5410 IF RI$ = "S" THEN GOTO 512
0
5420 END

```

Figura 6 - Esempio di schermata per la definizione degli input/output su disco del programma PCA. Si suppone l'esistenza di un file dati PCAFL. Verranno create le matrici, su dischetto e con il formato di figura 3: PCAFL.R (matrice delle correlazioni fra le variabili originarie); PCAFL.AVT (matrice degli autovettori associati alle componenti principali); PCAFL.S (matrice delle correlazioni fra le variabili originarie e le componenti principali). Il vettore PCA.AVL (degli autovalori estratti), invece, non verrà memorizzato.

File dati input: PCAFL

File di output  
(se si vuole la memorizzazione: S)

PCAFL.R	S
PCAFL.AVL	
PCAFL.AVT	S
PCAFL.S	S

lata l'inversa;

2) le variabili di dimensionamento devono avere i valori:

- C1 *autoregressione*: numero dei periodi pregressi + 2 (per i coefficienti  $b_1$  e  $b_2$ );

- C1 *multiregressione*: numero delle variabili, compresa la variabile "dummy" (colonna di tutti 1);

- R1 numero dei dati nella matrice A, pari al numero delle righe nel caso della multiregressione o  $< t-q$  nella autoregressione.

Si cancellano, quindi, le routine 10000 e 20000.

Questo dovrebbe bastare.

### Analisi delle componenti principali (PCA)

Le fondamenta dell'analisi delle componenti principali, (PCA), furono gettate da Pearson, nel 1901, come metodo per calcolare l'adattamento ottimale di

ellissoidi (o iperellissoidi) costituiti da punti in uno spazio  $q$ -dimensionale ad assi e piani ottenuti minimizzando la somma quadratica delle perpendicolari dai punti agli assi di riferimento.

Solo con i lavori di Fisher ed Hotelling, degli anni trenta, furono, però, formalizzate le metodologie e riconosciuti i sensi interpretativi delle tecniche conosciute oggi sotto il nome di PCA.

Ben più di recente è invece venuto l'uso generalizzato nella ricerca, soprattutto biologica e sociale, in tutti quei casi in cui la complessità del modello rende difficile la trattazione e l'analisi. Una delle definizioni più sintetiche e significative di questo tipo di analisi e dei suoi scopi è quella di Rao (1964):

"Quando è disponibile un gran numero di rilievi, è lecito domandarsi se sia possibile sostituirli con un insieme (da esso derivato) ridotto o con le loro funzioni, senza una grande perdita di informazione ... le componenti principali, che

sono funzioni lineari (dei rilievi) sono adatte a questo scopo".

La PCA, quindi, serve a rappresentare uno spazio  $q$ -dimensionale in un subspazio a  $p$  dimensione ( $p < q$ ) con la minor perdita di informazioni possibili.

La riduzione delle dimensioni del modello permette rappresentazioni, altrimenti impossibili, che anche nelle due o tre dimensioni, rispecchiano interrelazioni fra i dati rilevati che non sarebbero accessibili all'indagine, in termini di sole due o tre delle variabili originarie. Queste ultime, inoltre, possono essere messe in relazione con le componenti principali (o assi principali) in modo da poter ridurre il numero, operando una scelta non casuale.

Data una matrice  $A$  dove le colonne rappresentano le  $q$  variabili e le righe gli  $n$  individui, vogliamo compiere una trasformazione lineare delle variabili originali per ottenerne di nuove, ortogonali fra loro (cioè non correlate), con la ma-

#### Seguito listato PCA

```

10000 REM *****
10010 REM CARICA DATI
10020 REM *****
10030 REM
10040 PRINT D$;"OPEN";N$
10050 PRINT D$;"READ";N$
10060 INPUT ND: INPUT NC: INPUT
      NR
10070 IF NC > NR THEN GOSUB 500
      50
10080 DIM A(NR,NC + 1)
10090 FOR I = 1 TO NC
10100 FOR J = 1 TO NR
10110 INPUT A(J, I)
10120 NEXT : NEXT
10130 PRINT D$;"CLOSE"
10140 RETURN
20000 REM *****
20010 REM SCRIVI FILES
20020 REM *****
20030 REM
20040 PRINT D$;"OPEN";N$;".R"
20050 PRINT D$;"WRITE";N$;".R"
20060 PRINT 2: PRINT NC: PRINT N
      C
20070 FOR I = 1 TO NC
20080 FOR J = 1 TO NC
20090 PRINT R(J, I)
20100 NEXT : NEXT

```

```

20110 PRINT D$;"CLOSE"
20120 RETURN
20130 PRINT D$;"OPEN";N$;".VT"
20140 PRINT D$;"WRITE";N$;".VT"
20150 PRINT 2: PRINT NC: PRINT N
      C
20160 FOR I = NC TO 1 STEP - 1
20170 FOR J = 1 TO NC
20180 PRINT VT(J, I)
20190 NEXT : NEXT
20200 PRINT D$;"CLOSE"
20210 RETURN
20220 PRINT D$;"OPEN";N$;".VL"
20230 PRINT D$;"WRITE";N$;".VL"
20240 PRINT 2: PRINT 1: PRINT NC
20250 FOR I = NC TO 1 STEP - 1
20260 PRINT VL(I)
20270 NEXT
20280 PRINT D$;"CLOSE"
20290 RETURN
20300 PRINT D$;"OPEN";N$;".S"
20310 PRINT D$;"WRITE";N$;".S"
20320 PRINT 2: PRINT NC: PRINT N
      C
20330 FOR I = NC TO 1 STEP - 1
20340 FOR J = 1 TO NC
20350 PRINT S(J, I)
20360 NEXT : NEXT
20370 PRINT D$;"CLOSE"
20380 RETURN

```

### Seguito listato PCA

```

50000 REM *****
50010 REM ERROR MSG
50020 REM *****
50030 REM
50040 PRINT : PRINT "*** ROUT VT
      & VL ***": PRINT "IL RISULT
      ATO NON CONVERGE - ABORT": END

50050 PRINT : PRINT "*** ROUT CA
      RICA DATI ***": PRINT "I CAM
      PIONI SONO INSUFFICIENTI - A
      BORT": END

```

### Listato 3 - L'utility Edit.

```

1 REM *****
2 REM * EDIT PROGRAM *
3 REM * REL 1.1 *
4 REM * ----- *
5 REM * BY R.BRUNIALTI *
6 REM * ON 25/08/1982 *
7 REM *****
8 REM
9 REM
100 POKE 34,0: HOME
110 D$ = CHR$ (4)
120 HOME
130 VTAB 1: PRINT "NOME FILE";: HTAB
15: INPUT "":N$
140 HTAB 1
150 INPUT "RIGHE=";ND(1)
160 INPUT "COLONNE=";ND(2)
170 PRINT
180 POKE 34,4
190 VTAB 4
200 PRINT "-----"
      "-----"
210 DIM A(ND(1))
220 ND = ND(2)
230 XC = XC + 1
240 HTAB 20: VTAB 4: PRINT " COL
      ";XC;" ": PRINT
250 A = A + 1
260 IF A = ND(1) THEN PRINT CHR$
      (7);: HTAB 12: PRINT "(-- LA
      ST ITEM";: HTAB 1
270 PRINT A;");: HTAB 5: INPUT
      "":A$
280 IF A = ND(1) AND (A$ < ) "F
      " OR A$ < ) "E") THEN A(A) =
      VAL (A$):VC = PEEK (37): GOTO
      320

```

```

290 IF A$ = "F" OR A$ = "E" THEN
      VC = PEEK (37): GOTO 320
300 A(A) = VAL (A$)
310 GOTO 250
320 REM *****
330 REM INTERRUPT
340 REM *****
350 REM
360 IF A$ = "E" GOTO 450
370 A(A) = VAL (A$)
380 IF FL = 1 THEN 420
390 ND = ND - 1: GOSUB 570
400 IF ND < = 0 THEN 740
410 GOTO 230
420 ND = ND - 1: GOSUB 660
430 IF ND < = 0 THEN 740
440 GOTO 230
450 REM *****
460 REM TYPE ERROR
470 REM *****
480 REM
490 VTAB 10: HTAB 20
500 INPUT "N DATO ERRATO:";NN
510 HTAB 20
520 PRINT NN;");: INPUT A(NN)
530 HTAB 20
540 PRINT "ANCORA?";: GET Y$: IF
      Y$ = "N" THEN VTAB 10: HTAB
      20: PRINT "
      ": HTAB 20: PRINT "
      ": HTAB 20: PRINT
      "
      ":A = A -
      1: VTAB VC: GOTO 310
550 HTAB 20
560 VTAB 10: PRINT "
      ":
      HTAB 20: PRINT "
      ": HTAB
      20: PRINT "
      ": GOTO
      460
570 REM *****
580 REM SCRIVI COLONNA
590 REM *****
600 REM
610 FL = 1
620 PRINT D$;"OPEN ";N$
630 PRINT D$;"WRITE ";N$
640 PRINT 2
650 PRINT ND(2): PRINT ND(1)
660 IF FL THEN PRINT D$;"APPEND
      ";N$: PRINT D$;"WRITE ";N$
670 FOR I = 1 TO A
680 PRINT A(I)
690 NEXT
700 PRINT D$;"CLOSE"
710 A = 0
720 HOME
730 RETURN
740 END

```

trice di dispersione contenente tutti zeri tranne nella diagonale principale. La somma dei quadrati di questi, rappresenta la variazione totale sui nuovi assi;

Si può dimostrare che i valori della diagonale sono gli autovalori non nulli della matrice di correlazione delle variabili originarie, ricavati dalla soluzione dell'equazione:

$$(9) \quad R - \lambda I = 0$$

$R$  = matrice di correlazione

$$r_{ij} = \frac{\sum_{k=1}^m (a_{ki} - \bar{a}_i)(a_{kj} - \bar{a}_j)}{\sqrt{\sum_{k=1}^m (a_{ki} - \bar{a}_i)^2} \sqrt{\sum_{k=1}^m (a_{kj} - \bar{a}_j)^2}}$$

mentre gli autovettori associati  $V_i$  tali che  $\sum_{i=1}^q v_{ij}^2 = 1$ , permettono la trasformazione delle coordinate dei punti  $x_i$  nel nuovo sistema di riferimento  $Y$  dove:

$$(10) \quad y_{ij} = \sum_{h=1}^q v_{hj} (a_{ih} - \bar{a}_h)$$

quali gli assi originali sono ruotati. A questo punto si deve correlare la componente  $i$ -esima principale con le variabili originarie. È possibile dimostrare che:

$$(11) \quad \underline{S}_i = \underline{V}_i \text{SQR}(\lambda_i) = \begin{pmatrix} s_{i1} \\ \vdots \\ s_{iq} \end{pmatrix}$$

è il vettore delle correlazioni binarie tra la  $i$ -esima componente e la variabile originaria  $j$  (con  $j = 1, 2, \dots, q$ ).

A questo punto "i giochi" sono fatti. Infatti sceglieremo quelle  $p$  variabili che sono maggiormente correlate ( $s_{ij}$  maggiore) con le componenti principali che estraggono una maggior quota della varianza totale.

L'utilità della PCA non si esaurisce qui. Molte delle analisi statistiche si possono infatti compiere non più sui dati originali,  $a_{ij}$ , ma sulle loro trasformazioni nel

dai punti-campione;  
**2)** rotazione degli assi di un certo angolo per rendere ortogonali i dati. Ma ora vediamo i vantaggi offerti da questa tecnica nella rappresentazione dei dati.

Qualora si sia nella situazione in cui non si abbiano elementi su cui condurre una indagine (caso 2, esposto all'inizio di questo articolo), si può essere interessati a vedere graficamente se esistono "raggruppamenti" interni dei dati, clusters, e/o se anche le variabili si riuniscono in clusters.

### L'utilizzo della PCA nella individuazione dei clusters

L'utilizzo dei risultati della PCA nelle tecniche di ordinamento può essere basata sia sulle matrici che definiscono correlazioni tra gli attributi come le matrici di correlazione o dispersione (analisi di tipo Q).

Conviene, in genere, condurre analisi di tipo R quando, come nel nostro caso, sono più numerosi gli individui delle variabili.

Altrimenti, per economia di calcolo o per specificità di intenti, si compirà una analisi di tipo Q.

Rao suggerisce (Use and interpretation of principal component analysis in applied research - Sanhkyia. A - 1964) l'uso della seguente trasformazione per l'individuazione di clusters (i clusters possono essere definiti, più rigorosamente, come gruppi aventi varianza interna inferiore a quella totale):

$$(12) \quad Y = X \underline{V}$$

dove  $\underline{V}$  è la matrice contenente gli autovettori estratti dalla matrice  $R$ , normalizzati in modo che la somma dei quadrati di ogni colonna sia pari a uno.

La prima colonna della matrice  $\underline{V}$  offre la migliore rappresentazione possibile in una dimensione, le prime due nelle due dimensioni e così via. Infatti la coordinata  $i$ -esima di un campione nel nuovo sistema di riferimento non è altro che la somma delle variabili originarie, pesata con i coefficienti dell' $i$ -esimo vettore associato alla  $i$ -esima componente principale, normalizzato, che rappresenta i coseni dell'angolo per cui bisogna ruotare gli assi originari per ottene-

N. ordine	L	L/Q	Cum
I	7.459	.339	.339
II	3.029	.138	.477
III	1.808	.082	.559
IV	1.423	.065	.624
V	1.083	.049	.673
VI	.915	.042	.714
VII	.900	.041	.755
VIII	.760	.035	.790
IX	.676	.031	.821
X	.618	.028	.849
XI	.557	.025	.874
XII	.466	.021	.895
XIII	.409	.019	.914
XIV	.376	.017	.931
XV	.320	.015	.945
XVI	.290	.013	.959
XVII	.279	.013	.971
XVIII	.216	.010	.981
XIX	.165	7E-3	.989
XX	.105	5E-3	.993
XXI	.080	4E-3	.997
XXII	.065	3E-3	

Figura 7 - Output numerico, su schermo delle componenti principali estratte. Vengono presentati: i valori assoluti  $L$ , la % della varianza totale del modello spiegata dalla  $i$ -esima c.p.  $L/Q$ , e la % cumulata estratta dalle prime c.p. Come si vede le prime 7 (sulle 22 totali) estraggono ben il 75% della variazione totale del modello, quanto basta per un uso del modello semplificato a sole 7 delle variabili. I valori sono quelli, reali, di un caso pratico, che ci ha indotto alla scrittura del programma presentato.

$i = 1, 2, \dots, q$   
 $j = 1, 2, \dots, n$

Il rapporto  $\lambda_i / (\sum_{i=1}^q \lambda_i)$  esprime la percentuale della varianza totale estratta dalla  $i$ -esima componente principale.

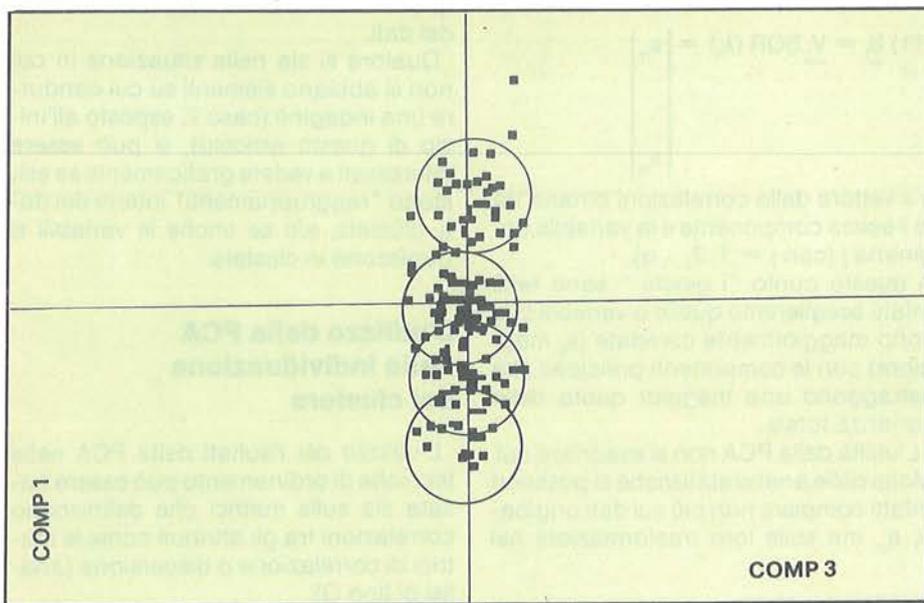
Geometricamente gli autovettori rappresentano i coseni degli angoli per i

sistema di riferimento  $y$ , con tutti i vantaggi che offrono variabili ortogonali. È però più difficile interpretare i risultati ed estenderne il significato sulle variabili originarie.

Si ricorda che, geometricamente, quello che si è compiuto è:

**1)** traslazione dell'origine degli assi al centro di massa del sistema costituito

Figura 8 - Esempio di output grafico di un caso reale, dove il modello comprendeva 22 variabili e 172 campioni. Si aveva modo di presumere l'esistenza di quattro raggruppamenti interni, clusters, che sono quelli individuati dai cerchi. La rappresentazione è stata fatta nello spazio della 1' e 3' c.p. . Come si vede nella realtà le situazioni non sono mai chiarissime. I dati sono gli stessi della figura 7.



re gli assi principali nell' (iper)spazio originale. La scelta dei due primi assi principali per la rappresentazione grafica dei punti campione non garantisce che un'apparente mancanza di addensamento dei punti sia, però, dovuta ad una effettiva mancanza di clusters.

Una diversa coppia (per esempio la prima e la terza) potrebbe rivelarne l'esistenza.

Dato il diverso peso dei nuovi assi come descrittori della dispersione originaria dei campioni (lasciata immutata, con tutte le sue proprietà geometriche, dalla trasformazione effettuata) non è conveniente spingersi, nella ricerca di clusters, oltre alle primissime componenti.

### Il programma PCA

Tutti i dati numerici che verranno usati in input ed in output, saranno letti e/o scritti su/da matrici aventi il formato di figura 3, come del resto per il programma Multireg.

Le fasi dell'analisi vengono compiute



## NEL PROSSIMO NUMERO TROVERETE

IN  
EDICOLA  
DAL  
26  
AGOSTO

**P**ROGETTO  
AEROMODELLO  
CON UN MSX

**S**COPA  
A TRE CARTE  
CON L'APPLE

**C**ARTELLE  
PER TUTTI  
CON LO  
SPECTRUM

**L**A GESTIONE  
DEL VIDEO  
CON LO  
SPECTRUM

**S**IMULAZIONE  
DINAMICA  
DI FENOMENI  
CONTINUI  
CON IL C 64

**D**IEL DATABASE  
SYNTHESIZER

## APPLE II

### UNA COMODA UTILITY: UN EDITOR PER MATRICI

Poiché si stanno trattando strutture di dati sotto forma, per lo più, di matrici o vettori, si è pensato fosse comodo disporre di un programma che permettesse la scrittura (ed eventuale correzione), su dischetto per Apple II DOS 3.3, delle stesse.

Il programma Edit soddisfa a queste esigenze.

Per una convenzione che ormai è entrata nell'uso di molti programmi di elaborazione dati numerici, le matrici vengono scritte facendo loro premettere una label di identificazione con tre valori, che determinano, rispettivamente, il numero delle dimensioni, il numero delle colonne e quello delle righe. Le dimensioni sono considerate sempre uguali a 2 (caso di una matrice) per permettere il prodotto matrice-vettore secondo le convenzioni dell'algebra matriciale. Altri valori, vengono usati per casi particolari (matrici alfanumeriche, matrici multidimensionali, ecc.). La label permette inoltre il dimensionamento automatico delle matrici nei programmi che le leggono.

*Si ricorda che nei programmi di statistica presentati, si è usata la convenzione di far corrispondere le colonne alle variabili e le righe ai profili individuali dei campioni rilevati.*

In fase editing c'è possibilità di correggere un eventuale errore battendo il carattere alfabetico E al posto del numero.

In questo caso verrà attivata una routine di correzione che chiederà il numero d'ordine del dato errato, che potrà quindi essere ribattuto correttamente. Se poi non ci sono altri errori da correggere (rispondere N alla domanda "Ancora?") si continua da dove si era interrotto.

Viene anche segnalato l'ultimo numero di ogni colonna (con la scritta "<— Last Item") dopo il quale verrà registrata sul dischetto tutta la colonna corrente di input. Dopo aver pulito lo schermo, si passerà all'input di un'altra colonna, sino al termine.

### Bibliografia

- 1) Costantini D. 1970, "Fondamenti del calcolo delle probabilità", Feltrinelli.
- 2) Sadocchi S. 1980, "Manuale di analisi statistica multivariata", F. Angeli.
- 3) Hudson J. 1983, "Linear regression", Practical Computer, Febbraio.
- 4) Rao C.R. 1968, "Use and interpretation of principal component analysis in applied research", Sankhya, Ser. A.
- 5) Greenstadt S. 1960, "The determination of the characteristic root of a matrix by the Jacobi method", Math.Meth. for dig. comp. - Wiley. N.Y.

## SUPER OFFERTA DI LUGLIO DEL COMPUTER STUDIO

Via Santhià, 47 - 10154 Torino - tel. (011) 287366

### Commodore 64

Interfaccia Centronics stampante	70.000
Idem grafica	95.000
Fucile ottico con serie giochi	70.000
Copricomputer in plexiglass	11.000
Joystick normale 2 pulsanti	14.000
Joystick fuoco rapido Spectravideo II o simile	18.000
Joystick tipo bar in metallo con microswitch	60.000
Joystick competition PRO 5000 con microswitch	39.000
Registratore dedicato	49.000
Penna ottica con software	52.000
Cassetta 10 giochi-novità	16.000
Idem disco	18.000

### Spectrum 16/48K

Tastiera professionale 16/48K	80.000
Kit espansione da 16 a 48K	49.000
Interfaccia Joystick singola	17.000
Interfaccia Joystick doppia	26.000
Interfaccia programmabile	52.000
Sintetizzatore vocale	70.000
Sintetizzatore musicale	85.000
Interfaccia RAM TURBO con ingresso	
2 Joystick più cartridge	70.000
Amplificatore suono con box	49.000
Gruppo 10 cassette gioco originali	28.000
Copricomputer in plexiglass Spectrum	10.000
Idem per Spectrum Plus	12.000
Idem per QL	15.000

### Commodore C16

Espansione di memoria 16K RAM	119.000
Gruppo 10 cassette gioco	40.000
Adattatore joystick	7.500
Adattatore registratore	7.500
Copricomputer in plexiglass	11.000

### Commodore VIC 20

Motherboard 4 ingressi per cartridge/espansioni	20.000
Cartridge basic extender	30.000
Espansione 3K grafica Commodore	30.000
Espansione 8K RAM	30.000
Espansione 3/8/16K RAM commut.	75.000
Espansione 32K commutabile	99.000
Interf. per tutti i reg. a cassette	12.000
Interfaccia stampante Centronics	70.000
Idem grafica	95.000
Penna ottica con software	52.000
Copricomputer in plexiglass	11.000
Gruppo 10 cassette gioco originali	20.000

### Altre offerte

10 Diskette di qualità SS/DD con box in plexiglass omaggio	35.500
Idem doppia faccia doppia densità DS/DD	53.000
Novità: accessorio tagliadischetti "Clipper" per utilizzare i dischetti singola faccia sui 2 lati	8.000
Set di pulizia testine e taratura cinghie registratore Commodore e altri	9.000
Gruppo 5 cassette C15 in vaschetta porta-cassette omaggio	5.000
Idem 4 confezioni (20 cassette C15 con relative vaschette)	14.500
Contenitore per 10 dischetti in plexiglass	3.500
Vaschetta porta dischetti floppy - coperchio in plexiglass con serratura e cerniera - capacità 50 pezzi	24.000
Idem - capacità 90 pezzi	29.000
Accessorio "stop joy" per bloccare al tavolo il tuo Joystick	7.500
Kit di taratura per azimuth dei registratori per Commodore 64 con manuale, software e cacciavite a croce	30.000

Vendita diretta o per corrispondenza. Ordini con pagamento anticipato: aggiungere L. 3.000 per spese postali. Ordini con pagamento contrassegno: inviare acconto L. 10.000. Le spese postali verranno addebitate al costo.

dal programma PCA. Questo legge (dal file <n.file>, richiesto dal programma stesso) quindi calcola la matrice  $R$  e ne estrae gli autovalori  $\lambda_i$  e gli autovettori associati. Questi ultimi li normalizza come descritto in precedenza.

I risultati possono essere scritti su file di nome:

```
<n.file> .R
<n.file> .AUVL
<n.file> .AUVT
```

per un eventuale utilizzo successivo. Quindi vengono mostrati i valori assoluti, relativi e cumulati di  $\lambda_i$ .

Si ricorda che gli autovalori vengono estratti in ordine decrescente di importanza. Essendo la matrice  $R$  simmetrica e positiva, è stato usato l'algoritmo ciclico di Jacobi.

Un ulteriore passo viene compiuto con il calcolo della matrice  $S$  delle correlazioni fra le variabili originarie e le componenti principali estratte. Il risultato può essere memorizzato nella matrice <n.file> .S, su dischetto.

Viene quindi domandato se si vogliono disegnare i punti-campione nello spazio di due delle p.c. estratte. Se si risponde affermativamente, comparirà il grafico con tutti gli n campioni.

### Modifiche del programma PCA per l'uso su altri computer

Il programma fa uso intensivo di comandi DOS 3.3 dell'Apple II. Per facilitare una conversione per altri computer, essi sono stati tutti raccolti nelle subroutine con numero di riga uguale o maggiore di 10000. Sostituendole adeguatamente, per un input manuale dei dati o per l'uso con altri sistemi operativi, non dovrebbero esserci problemi. Per facilitare la conversione sono stati scritti REMarks adeguati, specificando la funzione delle routine e i valori che le variabili devono assumere in entrata ed in uscita dalle stesse.

### Conclusioni

Si spera di aver svolto l'argomento in maniera rigorosa ma non (troppo) noiosa. Riteniamo, del resto, che solo un uso pratico dei programmi presentati possa far apprezzare la potenza dei mezzi di analisi e la comodità (speriamo) dei programmi stessi.

Questi sono stati scritti senza risparmio di Rem, anche a costo di allungarli un poco, al fine di facilitare la lettura e la comprensione degli algoritmi usati.

### REMARKS del programma PCA

Le linee contrassegnate con (\*) contengono istruzioni del S.O. dell'Apple II o istruzioni sue proprie.

**10-40** - Prepara lo schermo.

**\*1000-1260** - Input nome del file dati e scelta dei file di output.

**1270** - Richiamo routine di caricamento dati (10000).

**1280** - Dimensionamento matrici e vettori. I nomi sono autoesplicativi.

**2000-2100** - Calcolo medie e varianze delle variabili.

**2105-2140** - Standardizzazione dei dati ( $a_{ij} = (a_{ij} - a_i)/\text{sqr}(s_j)$ ).

**2150-2210** - Calcolo della matrice R delle correlazioni binarie delle variabili e stampa su dischetto, se selezionato nelle 1000-1260.

**3000-3530** - Calcolo degli autovettori ed autovalori della matrice R, secondo il metodo ciclico di Jacobi.

**3535-3555** - Stampa su video delle componenti principali (in valore assoluto, percentuale e cumulato).

**3560** - Se selezionata l'opzione nelle 1000-1260, richiamo delle subroutine di scrittura matrice degli autovettori su dischetto.

**3565** - Come sopra per il vettore degli autovalori.

**4000-4080** - Calcola la matrice, S, delle correlazioni fra le variabili originarie (righe) e le componenti principali estratte (colonne).

**4090** - Se selezionata l'opzione nelle 1000-1260, chiama la subroutine di scrittura matrice S, sempre su dischetto.

**\*5000-5070** - Inizializzazione delle variabili per la ricerca dei minimi e dei massimi, per lo scaling automatico dei plot.

**5080-5160** - Selezione, a cura dell'utente, delle componenti principali maggiormente significative.

**5170-5270** - Calcolo delle ascisse ed ordinate nel nuovo sistema di rappresentazione, nonché dei minimi e dei massimi dei due assi.

**5280-5310** - Ricerca dei valori per lo scaling e per il plottaggio.

**\*5320-5340** - Disegno assi e contorni del grafico.

**5350-5390** - Plottaggio dei punti campione.

**5400-5420** - Richiesta di eventuale nuovo plottaggio.

**\*10000** - Routine di caricamento dati. Vengono assegnati i valori NC = numero colonne matrice, NR = numero righe e dimensionata la matrice A (NR, NC + 1).

**\*20000** - Routine per la scrittura dei file selezionati.

**50000-50190** - Messaggi di errore, che provocano la interruzione del programma.

**Nota** - La matrice VT ha le colonne invertite (il primo autovettore è nella ultima colonna). Similmente il vettore VL, dove il primo autovalore è nella posizione NC. ■

**Q**uesto semplice programma (non vi spaventate guardando il listato: la lunghezza è dovuta soprattutto alla serie di linee Data!) vuole dimostrare come sia possibile costruire qualcosa di interessante e di carino attorno ad una idea anche modesta.

La cosa è nata da una personale "simpatia" per il programma *Titoli RND*, pubblicato sul n. 33 di Bit (Novembre 1982 - rubrica Apple Club - pag. 145). A grandi linee, le funzioni svolte dal programma sono queste: legge una stringa contenuta in una linea Data; gli dà una bella "shakerata", anagrammandone le lettere; fa "cadere" le lettere ad una ad una su di un "piano" posto all'estremo inferiore dello schermo; qui giunte, le lettere vanno ad occupare ciascuna il proprio posto, ricomponendo e visualizzando a poco a poco la parola (o frase) originaria.

Bello vero! Così ci mettemmo in testa di tradurre il programma originale (per Apple) affinché potesse girare sullo Spectrum.

La cosa non risultò neppure così complessa, anche grazie alla potenza delle istruzioni di "string-slicing" che il nostro nero amico ci mette a disposizione.

Per chi ancora non lo sapesse, l'istruzione MID\$ del BASIC standard è stata trasformata nella A\$ = B\$(X TO Y), secondo noi molto più comoda e versatile. Esempio: se in Applesoft vogliamo estrarre i caratteri dal 2 al 4 della stringa A\$, useremo una B\$ = MID\$(A\$,2,3); con lo Spectrum più semplicemente B\$ = A\$(2 TO 4): visto? non occorre più ricordarsi se il primo carattere scelto è compreso, o no, o forse, ma allora, vediamo, boh!.....abbiamo reso l'idea?!

Una volta riusciti nell'impresa, stavamo rimirando il mirabolante (!) risultato, quando ci accorgemmo che un cuginetto di 8 anni, si divertiva molto a cercare di indovinare la parola prima che la sua "ricomposizione" fosse completa, (programmatori seri: se non avete un cugino a disposizione *procuratevelo!* vi darà idee meravigliose!!)

Ed ecco il lampo! Perché non elaborare un po' il tutto, aggiungere un po' di grafica qui, un po' di suoni là, e magari mettere il tutto sotto forma di gioco per due persone, tanto per rendere la cosa più appassionante?

E a questo punto, perché non unire il

di Federico Lo Cicero

# Geo-Race

Come una semplice idea può diventare un simpatico programma

dilettevole all'utile mettendo come Data qualcosa di "scolastico"?

Detto fatto, ecco disegnato il campo di gioco: un trampolino dal quale una bella ruspa fa cadere le lettere, un tappeto pneumatico per raccoglierle, un display che tiene i punti....et voilà il gioco (è proprio il caso di dirlo!) è fatto.

Come materia è stata scelta la geografia (da qui il nome del programma), e precisamente una accoppiata città-re-

sbaglia, perde punti e l'indicatore scende.

Vince naturalmente il giocatore che raggiunge per primo il "fondo-scala". Un'ultima nota di programmazione: Geo-Race è di quanto più anti-strutturale si possa trovare.

Noterete che ogni tanto una Inkey\$ intercetta la pressione di un tasto: molti di voi inorridiranno e chiederanno "mai sentito parlare di subroutine?!" Giusto,

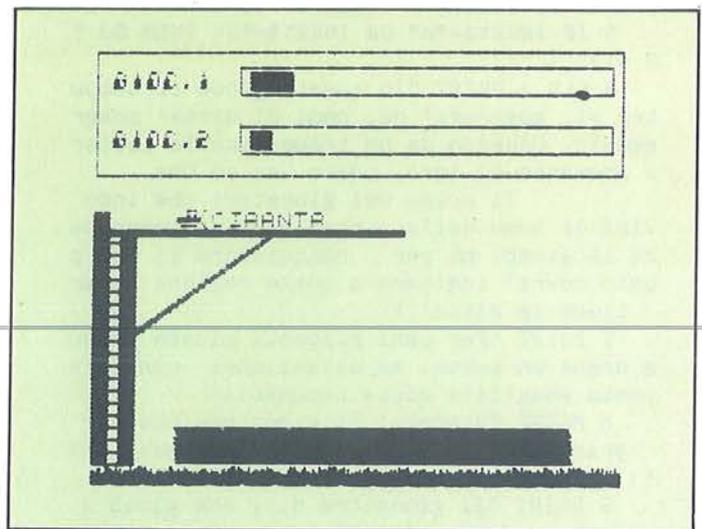


Figura 1 - Un hard copy del gioco.

gione di appartenenza.

Ma se preferite potrete cambiare facilmente materia, modificando solo le Data ed il "prompt" alla linea 68.

Troverete le istruzioni nel listato; comunque sappiate che si gioca in due: il primo dei giocatori che indovina la città, preme il suo bravo tasto e ferma il gioco. Gli viene chiesta la regione corrispondente e, se indovina, il suo punteggio aumenta e l'indicatore di punti sale; se

noi vi giriamo la domanda. "mai sentito parlare dei tempi di ricerca delle G-sub?", (voi potreste replicare suggerendo di piazzare la subroutine all'inizio del listato, ed io allora...ma non la finiremo più).

Quello che conta è che così il programma è sufficientemente veloce nel riconoscere la pressione di uno dei tasti di gioco, e questo è quello che importa. Il programma così com'è presentato gi-

ra sul 48 Kbyte, ma è facile farlo stare anche nel 16, accorciando la lista delle Data e magari registrando gli U.D.G. una volta per tutte come "Code".

Buon divertimento (a battere il listato voi, e a giocare i vostri pargoli, figli, fratellini o cuginetti che siano....! e se non ne avete a disposizione, procurateveli, come già accennato in precedenza).

## Le principali linee del programma

- 1-2 - Inizializzazione.
- 4-9 - Istruzioni.
- 12-25 - Preparazione del campo di gioco.
- 26 - Lettura delle linee Data.
- 27-54 - Routine principale di scomposizione e composizione delle parole.

56-62 - U.D.G.

73-84 - Richiesta risposta, riconoscimento della stessa, aggiornamento del punteggio e dei display.

84-179 - Data.

181 - Vittoria!

### Listato 1 - Il programma Geo-Race

```

10 BORDER 1: CLS
20 LOAD "pic"SCREEN$
30 PAPER 1: INK 1
40 LOAD "baratro"

1 DIM s(2): DIM x(2): LET x(1)=75: LET
x(2)=75: LET s(1)=0: LET s(2)=0
2 GO SUB 0056: REM UDG
3 BORDER 1: PAPER 1: INK 7: CLS
4 PRINT AT 0,0;"Vuoi le istruzioni (s/n)
)? ": IF INKEY$="" THEN GO TO 04
5 IF INKEY$="n" OR INKEY$="N" THEN GO T
O 012
6 CLS : PRINT "In questo gioco il compu
ter vi mostrera' dei nomi di citta' anagr
ammati. Cadendo da un trampolino le letter
e andranno al loro posto una ad una.
Il primo dei giocatori che indo
vina il nome della citta' puo' interrompe
re il gioco, ma per conquistare il suo p
unto dovra' indicare a quale regione appar
tiene la citta'."
7 PRINT "Per ogni risposta giusta si gu
a dagna un punto, ma attenzione: ogni ris
posta sbagliata costa un punto!!"
8 PRINT "Vincera' il giocatore che per
primo fornira' 17 risposte esatte!!
": PRINT
9 PRINT "Il giocatore n.1, che gioca a
destra, deve usare il tasto <O>": PRINT
: PRINT "Il giocatore n.2, che gioca a s
inistra, deve usare invece il tasto <I>."

10 PRINT : PRINT : PRINT
11 PRINT FLASH 1;AT 20,0;"PREMERE UN TA
STO PER CONTINUARE": PAUSE 0
12 BORDER 0: INK 5: PAPER 0: CLS : PLOT
0,8: DRAW 19,0: FOR y=8 TO 103: PLOT 0,y:
DRAW 7,0: PLOT 13,y: DRAW 7,0: NEXT y
13 FOR Y=104 TO 110: PLOT 0,y: DRAW 7,0:

```

```

NEXT y
14 FOR y=8 TO 103 STEP 5: PLOT 8,y: DRAW
5,0: NEXT y
15 FOR y=101 TO 103: PLOT 0,y: DRAW 150,
0: NEXT y
16 PLOT 20,60: DRAW 70,43
17 PLOT 20,61: DRAW 69,42: PLOT 20,62: D
RAW 68,41
18 INK 4: FOR x=0 TO 255: PLOT x,0: DRAW
0,RND*3+4: NEXT x
19 INK 2: FOR y=9 TO 23: PLOT RND*1+40,y
: DRAW RND*1+190,0: NEXT y
20 INK 6: PLOT 1,175: DRAW 254,0: DRAW 0
,-50: DRAW -254,0: DRAW 0,50
21 PRINT AT 1,1;"{C}{D}{E}{F}{G}{H}";AT
4,1;"{C}{D}{E}{F}{G}{I}"
22 PLOT 70,169: DRAW 180,0: DRAW 0,-11:
DRAW -180,0: DRAW 0,11
23 PLOT 70,145: DRAW 180,0: DRAW 0,-11:
DRAW -180,0: DRAW 0,12
24 INK 7: LET x1=75: LET n1=3: DIM a(40)
: DIM c(40)
25 PRINT INK 0;AT 8,1;"
";AT 18,4;"
":
INK 7
26 RANDOMIZE : RESTORE RND*95+85: LET k=
1: LET p1=19: READ a$: READ h$: GO SUB 003
2
27 FOR t=1 TO 200: LET k$=INKEY$: IF k$<
>" THEN LET p1=1160: GO TO 0063
28 NEXT t
29 FOR x=17 TO 1 STEP -1: PRINT AT 8,x;"
{A}{B}": FOR t=1 TO 2: BEEP .001,RND*10+10
: NEXT t: PRINT AT 8,x+1;" ": NEXT x: PRIN
T AT 8,1;" "
30 PRINT FLASH 1;AT 8,1;"premi un tasto"
: PAUSE 0
31 GO TO 0025
32 GO SUB 0038: GO SUB 0043: GO SUB 0047
: GO SUB 0051
33 LET pf=1: LET cb=0
34 FOR l=LEN (a$) TO 1 STEP -1

```

## Seguito programma Geo-Race

```

35 LET cb=cb+1: LET b$=r$(1+2): LET r$=r
$( TO 1+2): PRINT AT 8,pf-1;" ": PRINT AT
8,pf;r$
36 LET k$=INKEY$: IF k$<>" " THEN LET pl=
37: GO TO 0063
37 LET pf=pf+1: GO SUB 0052: PRINT AT pl
-1,c(1)+INT ((30-LEN (a$))/2);b$: NEXT l:
RETURN
38 FOR x=1 TO 31: LET a(x)=0: NEXT x: LE
T r$="": LET ci=0
39 LET ps=INT (RND*LEN (a$))+1: IF a(ps)
=1 THEN GO TO 0039
40 LET a(ps)=1: LET ci=ci+1: LET c(ci)=p
s: LET r$=r$+a$(ps)
41 IF ci<LEN (a$) THEN GO TO 0039
42 LET r$="{A}{B}" + r$: RETURN
43 FOR i=1 TO LEN (a$)
44 PRINT AT 8,i;r$(i+2): BEEP .001,RND*1
0+10
45 LET k$=INKEY$: IF k$<>" " THEN LET pl=
46: GO TO 0063
46 NEXT i: RETURN
47 FOR i=1 TO 19-LEN (r$)
48 PRINT AT 8,i;r$
49 LET k$=INKEY$: IF k$<>" " THEN LET pl=
50: GO TO 0063
50 BEEP .035,RND*10+10: PRINT AT 8,i;"
": NEXT i: RETURN
51 PRINT AT 8,i;r$: RETURN
52 FOR i=8 TO p1-2
53 LET k$=INKEY$: IF k$<>" " THEN LET pl=
54: GO TO 0063
54 PRINT AT i,19;b$: BEEP .035,RND*10+10
: PRINT AT i,19;" ": NEXT i: RETURN
56 RESTORE 0057: FOR q=97 TO 105: FOR n=
0 TO 7: READ a: POKE USR CHR$ (q)+n,a: NEX
T n: NEXT q
57 DATA 7,4,6,255,255,64,63,0,198,68,200
,240,240,40,196,6
58 DATA 28,34,32,118,114,114,124,0,8,8,8
,28,28,28,28,0
59 DATA 28,34,34,114,114,114,124,0,28,34
,32,112,112,114,124,0
60 DATA 0,0,0,0,0,24,24,0,8,24,8,28,28,2
8,28,0
61 DATA 60,2,2,124,112,112,126,0
62 RETURN
63 IF k$<>"0" AND k$<>"1" THEN GO TO pl
64 LET rg=VAL (k$)+1
65 GO TO ((rg=1)*66)+((rg=2)*67)
66 PRINT FLASH 1;AT 1,1;"{C}{D}{E}{F}{G}
{H}": GO SUB 0068: PRINT INK 6;AT 1,1;"{C}
{D}{E}{F}{G}{H}": GO TO 0025

```

```

67 PRINT FLASH 1;AT 4,1;"{C}{D}{E}{F}{G}
{I}": GO SUB 0068: PRINT INK 6;AT 4,1;"{C}
{D}{E}{F}{G}{I}": GO TO 0025
68 INPUT AT 0,0;"IN QUALE REGIONE SI TRO
VA ? "; LINE W$
69 FOR t=1 TO LEN (w$): IF CODE (w$(t))>
96 THEN LET w$(t)=CHR$ (CODE (w$(t))-32)
70 NEXT t
71 LET rv=(w$=h$): GO TO ((rv=1)*73)+((r
v=0)*78)
73 LET q$="BRAVO! E' ESATTO!": GO SUB 00
82
74 INK 2: FOR x=x(rg) TO x(rg)+10: PLOT
x,((rg=1)*160)+((rg=2)*136): DRAW 0,7: BEE
P .1,x/5: NEXT x: LET rv=0: LET x(rg)=x(rg)
+10: LET s(rg)=s(rg)+1
75 IF s(rg)=17 THEN GO TO 0180
76 RETURN
78 IF s(rg)=0 THEN LET rv=0: LET q$="SBA
GLIATO! RITENTA..": GO SUB 0082: RETURN
79 INK 0: FOR x=x(rg) TO x(rg)-10 STEP -
1: PLOT x,((rg=1)*160)+((rg=2)*136): DRAW
0,7: BEEP .1,x/5: NEXT x: LET rv=0: LET x(
rg)=x(rg)-10: LET s(rg)=s(rg)-1
80 LET q$="SBAGLIATO! RITENTA..": GO SUB
0082: RETURN
82 PRINT FLASH 1; INK 7;AT 15,(32-LEN (q
$))/2;q$
83 FOR t=1 TO 300: NEXT t: PRINT INK 0;A
T 15,3;" "
84 RETURN
85 DATA "TORINO","PIEMONTE"
86 DATA "ALESSANDRIA","PIEMONTE"
87 DATA "ASTI","PIEMONTE"
88 DATA "CUNEO","PIEMONTE"
89 DATA "NOVARA","PIEMONTE"
90 DATA "VERCELLI","PIEMONTE"
91 DATA "AOSTA","VALLE D'AOSTA"
92 DATA "MILANO","LOMBARDIA"
93 DATA "COMO","LOMBARDIA"
94 DATA "SONDRIO","LOMBARDIA"
95 DATA "VARESE","LOMBARDIA"
96 DATA "BERGAMO","LOMBARDIA"
97 DATA "BRESCIA","LOMBARDIA"
98 DATA "PAVIA","LOMBARDIA"
99 DATA "CREMONA","LOMBARDIA"
100 DATA "MANTOVA","LOMBARDIA"
101 DATA "GENOVA","LIGURIA"
102 DATA "SAVONA","LIGURIA"
103 DATA "IMPERIA","LIGURIA"
104 DATA "LA SPEZIA","LIGURIA"
105 DATA "VENEZIA","VENETO"
106 DATA "BELLUNO","VENETO"
107 DATA "PADOVA","VENETO"
108 DATA "ROVIGO","VENETO"

```

## Seguito programma Geo-Race

```

109 DATA "TREVISO","VENETO"
110 DATA "VERONA","VENETO"
111 DATA "VICENZA","VENETO"
112 DATA "TRENTO","TRENTINO ALTO ADIGE"
113 DATA "BOLZANO","TRENTINO ALTO ADIGE"
114 DATA "TRIESTE","FRIULI VENEZIA GIULIA"
"
115 DATA "UDINE","FRIULI VENEZIA GIULIA"
116 DATA "GORIZIA","FRIULI VENEZIA GIULIA"
"
117 DATA "PORDENONE","FRIULI VENEZIA GIULIA"
IA"
118 DATA "BOLOGNA","EMILIA ROMAGNA"
119 DATA "FERRARA","EMILIA ROMAGNA"
120 DATA "FORLI","EMILIA ROMAGNA"
121 DATA "MODENA","EMILIA ROMAGNA"
122 DATA "PARMA","EMILIA ROMAGNA"
123 DATA "PIACENZA","EMILIA ROMAGNA"
124 DATA "RAVENNA","EMILIA ROMAGNA"
125 DATA "REGGIO EMILIA","EMILIA ROMAGNA"

126 DATA "FIRENZE","TOSCANA"
127 DATA "AREZZO","TOSCANA"
128 DATA "GROSSETO","TOSCANA"
129 DATA "LIVORNO","TOSCANA"
130 DATA "LUCCA","TOSCANA"
131 DATA "MASSA CARRARA","TOSCANA"
132 DATA "PISA","TOSCANA"
133 DATA "PISTOIA","TOSCANA"
134 DATA "SIENA","TOSCANA"
135 DATA "ANCONA","MARCHE"
136 DATA "ASCOLI PICENO","MARCHE"
137 DATA "MACERATA","MARCHE"
138 DATA "PESARO","MARCHE"
139 DATA "PERUGIA","UMBRIA"
140 DATA "TERNI","UMBRIA"
141 DATA "ROMA","LAZIO"
142 DATA "FROSINONE","LAZIO"
143 DATA "LATINA","LAZIO"
144 DATA "RIETI","LAZIO"
145 DATA "VITERBO","LAZIO"
146 DATA "L'AQUILA","ABRUZZO"
147 DATA "CHIETI","ABRUZZO"
148 DATA "PESCARA","ABRUZZO"
149 DATA "TERAMO","ABRUZZO"
150 DATA "CAMPOBASSO","MOLISE"
151 DATA "ISERNIA","MOLISE"
152 DATA "NAPOLI","CAMPANIA"
153 DATA "AVELLINO","CAMPANIA"
154 DATA "BENEVENTO","CAMPANIA"
155 DATA "CASERTA","CAMPANIA"
156 DATA "SALERNO","CAMPANIA"
157 DATA "BARI","PUGLIA"

158 DATA "BRINDISI","PUGLIA"
159 DATA "FOGGIA","PUGLIA"
160 DATA "LECCE","PUGLIA"
161 DATA "TARANTO","PUGLIA"
162 DATA "POTENZA","BASILICATA"
163 DATA "MATERA","BASILICATA"
164 DATA "CATANZARO","CALABRIA"
165 DATA "COSENZA","CALABRIA"
166 DATA "REGGIO CALABRIA","CALABRIA"
167 DATA "PALERMO","SICILIA"
168 DATA "AGRIGENTO","SICILIA"
169 DATA "CALTANISSETTA","SICILIA"
170 DATA "CATANIA","SICILIA"
171 DATA "ENNA","SICILIA"
172 DATA "MESSINA","SICILIA"
173 DATA "RAGUSA","SICILIA"
174 DATA "SIRACUSA","SICILIA"
175 DATA "TRAPANI","SICILIA"
176 DATA "CAGLIARI","SARDEGNA"
177 DATA "SASSARI","SARDEGNA"
178 DATA "NUORO","SARDEGNA"
179 DATA "ORISTANO","SARDEGNA"
181 FOR t=1 TO 7: BEEP .1,10: NEXT t
182 PAPER 7: BORDER 7: INK 1: CLS
183 PRINT AT 1,0;"MOLTO BENE! IL GIOCATOR
E NUMERO ";r;g;" HA VINTO LA PARTITA!"
184 PRINT AT 4,0;"NE VOLETE FARE UN'ALTRA
(S/N) ?"
185 IF INKEY$="s" THEN RUN
186 IF INKEY$="n" THEN STOP
187 GO TO 0184

```

# Routine 2 per Spectrum

Una copiosa raccolta di effetti  
per il vostro computer

di Ivano Parbuono

Come già descritto nella prima raccolta di *Routine* in precedenza pubblicata (**Personal Software** n. 22), dalla necessità appunto di avere sempre a portata di mano una serie di effetti particolari e piccole mini-routine dalla facile consultazione, abbiamo creduto opportuno creare questa seconda raccolta che differisce completamente dalla prima.

In alcuni casi si va ad agire per mezzo dei Data e dei Poke direttamente nelle locazioni di memoria del computer creando così quegli effetti che sono propri del linguaggio macchina e che la maggior parte delle volte era possibile creare solamente caricandoli da cassette preregistrate.

Iniziamo l'analisi dalla riga 100 ove comincia una routine di "scritte in rilievo con alternanza di colori".

Naturalmente si può scrivere tutto quello che si desidera, in questo caso come in altre routine successive è stato messo il nome dell'autore ma solo a carattere puramente indicativo. Alla li-

nea 120 inizia un ciclo che termina alla linea 185 e permette alle scritte realizzate attraverso i Plot e Draw di spostarsi a destra da 1 a 8 con un salto di 1,7 ed ogni volta realizzato con un colore diverso dando così la sensazione di una scritta in rilievo. Dalla linea 192 alla linea

280 continua la seconda parte della scritta in rilievo riempiendo così tutto lo schermo per dare poi la dimostrazione della seconda routine che inizia alla linea 350 e termina alla linea 430. Ma vediamo ora linea per linea anche questa routine, che crea lo scroll a sinistra

Listato 1 - Il programma con tutte le routine presentate.

```

80 REM *****
82 REM *****
84 REM ** R O U T I N E 2 **
86 REM *****
88 REM *****
100 REM scritte in rilievo con
alternanza di colori
120 BORDER 6: FOR P=1 TO 8 STEP
1.7: INK (RND*6): PAUSE 5
150 PLOT 24+P,136-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-12: DRAW -16
,0: DRAW 16,-12
155 PLOT 66+P,136-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-24: DRAW -16
,0
160 PLOT 88+P,160-P: DRAW 0,-24
: DRAW 16,0: DRAW 0,24
165 PLOT 122+P,160-P: DRAW 26,0
: PLOT 136+P,160-P: DRAW 0,-24
170 PLOT 166+P,160-P: DRAW 0,-2
4
175 PLOT 184+P,136-P: DRAW 0,24
: DRAW 16,-24: DRAW 0,24
180 PLOT 232+P,136-P: DRAW -16,
0: DRAW 0,12: DRAW 10,0: DRAW -1
0,0: DRAW 0,12: DRAW 16,0
185 NEXT P
190 INK 2: PRINT AT 11,2: FLASH
1: BRIGHT 1: "REALIZZATA DA IVAN
O PARBUONO": FLASH 0
192 REM seconda parte del video
, segue scritte in rilievo con al
ternanza di colori
195 FOR P=1 TO 6 STEP 1.6: INK

```

```

(RND*6): PAUSE 5
200 PLOT 8+P,16-P: DRAW 0,16
205 PLOT 15+P,32-P: DRAW 4,-16:
DRAW 4,16
210 PLOT 31+P,16-P: DRAW 4,16:
DRAW 4,-16: PLOT 35+P,24-P: DRAW
4,0
215 PLOT 48+P,16-P: DRAW 0,16:
DRAW 8,-16: DRAW 0,16
220 PLOT 64+P,16-P: DRAW 0,16:
DRAW 4,0: DRAW 0,-16: DRAW -4,0
225 PLOT 128+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-8: DRAW -8,0
230 PLOT 144+P,16-P: DRAW 4,16:
DRAW 4,-16: PLOT 148+P,24-P: DR
AW 4,0
235 PLOT 160+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-8: DRAW -8,0:
DRAW 8,-8
240 PLOT 176+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-6: DRAW -3,-2
: DRAW -5,0: DRAW 5,0: DRAW 3,-2
: DRAW 0,-6: DRAW -8,0
245 PLOT 192+P,32-P: DRAW 0,-16
: DRAW 8,0: DRAW 0,16
250 PLOT 208+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-16: DRAW -8,0
255 PLOT 224+P,16-P: DRAW 0,16:
DRAW 8,-16: DRAW 0,16
260 PLOT 240+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-16: DRAW -8,0
280 NEXT P
300 PAUSE 200
350 REM routine per lo SCROLL a
sinistra con rigenerazione di c
aratteri

```

con rigenerazione continua di caratteri. La linea 390 con un Gosub ci manda alla 410 dove troviamo un Restore che va a leggere i Data di linea 430 per tornare poi alla seconda parte di linea 410 dove il For R = 0 To 17 colloca per mezzo del Poke 61000 alle locazioni successive i 17 Data in precedenza letti e dopo ciò per mezzo di un Return ritor-

na alla linea 400 dove il ciclo presente permette per 56 volte lo scroll verso sinistra con la rigenerazione dei caratteri intervallata da un Beep e un Pause in funzione Rnd. L'istruzione Let A = Usr 61000 ha il compito di leggere tutti i Data registrati in precedenza alle locazioni successive alla 61000 sino a trovare il Data 201 che corrisponde al Return.

L'effetto di movimento in alto con sonoro si può ottenere con la routine che inizia alla linea 500 con assegnazione di valori alla linea 510 ed inizio del loop per il riempimento totale del video alla 520. Alla linea 530 si può scrivere ciò che si desidera, mentre la linea 540 cambia ogni volta i valori in precedenza assegnati per terminare poi con un Beep del

## Seguito listato 1.

```

390 GO SUB 410
400 FOR I=1 TO 56: BEEP .05, (RN
D*8): LET A=USR 61000: PAUSE (RN
D*10)+1: NEXT I
405 PAUSE 200: CLS : PAPER 7: G
O TO 500
410 RESTORE 430: FOR R=0 TO 17:
READ code: POKE 61000+R,code: N
EXT R: RETURN
430 DATA 33,255,87,14,32,167,20
3,6,43,13,32,250,62,63,166,32,24
2,201
500 REM effetto movimento in al
to con sonoro
510 LET P=2: LET I=7
520 FOR S=0 TO 21
530 PRINT AT S,8: INK P: PAPER
I: FLASH 1:"PARBUONO IVANO"
540 LET V=P: LET P=I: LET I=V:
NEXT S
550 FOR T=0 TO 25: BEEP 0.5,3
560 NEXT T
570 PAUSE 200: CLS : GO TO 600
600 REM creazione di un grafico
per cornice
610 INK 0: FOR I=PI-PI TO VAL "
248" STEP VAL "8": PLOT I,VAL "1
60": DRAW VAL "7",VAL "7": PLOT
I,VAL "167": DRAW VAL "7",VAL "
7": NEXT I
620 FOR I=PI-PI TO VAL "248" ST
EP VAL "8": PLOT I,VAL "0": DRAW
VAL "7",VAL "7": PLOT I,VAL "7"
: DRAW VAL "7",VAL "7": NEXT I
630 PAUSE 200: PRINT AT 12,5:"C
ORNICE PER DOMANDE"
650 PAUSE 100: BORDER 7: PAPER
7: CLS : GO TO 700
700 REM routine per la creazion
e dell'effetto terremoto
705 PRINT AT 21,3:"EFFETTO TERR
EMOTO"
710 PLOT 24,88: DRAW 55,46: DRA
W 56,-46: PLOT 24,88: DRAW 56,40
: DRAW 56,-40: INK 3: PLOT 48,10
4: DRAW 65,0: DRAW 0,-88: DRAW -
65,0: DRAW 0,88
712 PLOT 72,16: DRAW 0,32: DRAW
15,0: DRAW 0,-32
714 PLOT 32,14: DRAW 136,0: DRA
W 5,-10: DRAW 48,0: PLOT 56,86:
DRAW 13,0: DRAW 0,-20: DRAW -13,
0: DRAW 0,20: PLOT 103,88: DRAW
-13,0: DRAW 0,-20: DRAW 13,0: DR
AW 0,20
716 PRINT AT 20,24:"■":AT 19,24

```

```

"■":AT 18,24:"■":AT 17,24:"■":A
T 16,24:"■":AT 15,24:"■":AT 14,2
4:"■":AT 13,24:"■":AT 15,23:"■":
AT 14,22:"■":AT 14,25:"■":AT 13,
26:"■":AT 12,23:"■":AT 12,25:"■"
718 PRINT AT 11,23:"|":AT 11,25
:"|":AT 10,22:"|":AT 10,26:"|"
720 GO SUB 770
760 FOR I=1 TO 25: LET A=USR 61
100: PAUSE (RND*10)+1: NEXT I
765 PAUSE 200: CLS : GO TO 800
770 RESTORE 780: FOR P=0 TO 76:
READ code: POKE 61100+P,code: N
EXT P: RETURN
780 DATA 58,0,64,246,58,255,90,
245,58,255,87,245,33,0,64,84,93,
1,254,226,35,237,176,251,118,118,
118,118,58,255,87,50,254,87,33,2
55,90,84,93,1,253,26,43,43
785 DATA 237,184,251,118,118,11
8,118,118,118,33,0,64,84,93,1,25
4,26,35,237,176,241,50,255,87,24
1,50,255,90,241,50,0,64,201
2010 BORDER 7: PAPER 7
2015 REM creazione della scritta
ROUTINE in rilievo
2020 FOR P=1 TO 8: PAUSE 10
2030 PLOT 16+P,72-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-12: DRAW -16,
0: DRAW 16,-12
2040 PLOT 48+P,72-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-24: DRAW -16,
0
2050 PLOT 80+P,95-P: DRAW 0,-24:
DRAW 16,0: DRAW 0,24
2060 PLOT 115+P,95-P: DRAW 26,0:
PLOT 128+P,95-P: DRAW 0,-24
2070 PLOT 160+P,95-P: DRAW 0,-24
2080 PLOT 176+P,72-P: DRAW 0,24:
DRAW 16,-24: DRAW 0,24
2090 PLOT 224+P,72-P: DRAW -16,0
: DRAW 0,12: DRAW 16,0: DRAW -10
0: DRAW 0,12: DRAW 16,0
2095 NEXT P: GO SUB 2110
2100 REM routine per la creazion
e dello SCROLL LATERALE CON RIEN
TRO NEGATIVO E POSITIVO
2102 FOR I=1 TO 512: LET A=USR 6
1200: NEXT I
2105 PAUSE 200: GO TO 2200
2110 RESTORE 2130: FOR R=0 TO 31
: READ code: POKE 61200+R,code:
NEXT R: RETURN
2130 DATA 33,255,87,6,192,197,22
9,221,225,221,126,225,7,0,63,6,3
2,203,22,43,16,251,193,16,236,20
1,241,43,193,16,230,201
2310 GO SUB 2330

```

**E' IN EDICOLA**

# **ELETTRONICA 1**

Mensile di elettronica pratica e hardware per microcomputer

L. 3.000

MAGGIO 85

GRUPPO  
EDITORIALE  
JACKSON

Nuova edizione della rivista

Speciale  
**ANTIFURTO  
DIGITALE**

Spedizione in Abb. Post. Gruppo n. 70

**30W E PIU'  
D' USCITA  
UNITA PORTATILE  
PER CHITARRA**

**COMPUTERJOYSTICK  
ANALOGICO**

**MOLTO PIU'  
DI UNA RIVISTA  
DI ELETTRONICA**



**GRUPPO  
EDITORIALE  
JACKSON**

Milano-San Francisco-Londra-Madrid

loop di linea 550. Creazione di un grafico da usare tipo cornice: questa routine che va da linea 600 e termina alla linea 650 permette appunto la creazione di due linee intrecciate ad altezza ed ampiezza desiderata sullo schermo; si può modificare l'altezza, la lunghezza e l'ampiezza intervenendo alla linea 610. In ogni caso qualsiasi dato venga cam-

biato crea una cornice totalmente diversa. Questo discorso vale anche per la linea 620 che completa questa piccola routine. Alla linea 700 inizia un'altra routine per creare "l'effetto terremoto", alla linea 710 è stato assegnato un grafico da rappresentare che servirà per la dimostrazione di questa routine. Alla linea 720 è stato assegnato un Gosub 770 che

va a leggere i 76 Data successivi, quindi troviamo il ciclo che assegna i Data in precedenza letti alle locazioni che vanno dalla 61100 alla 61176. Terminata l'assegnazione di questi dati in memoria, il programma ci rimanda alla linea 760 ove un ulteriore ciclo va a leggere in memoria tutti i 76 Data in precedenza memorizzati creando così l'effetto che

Seguito listato 1.

```

2315 REM routine per la creazione
e dello SCROLL VELOCE A SINISTRA
CON RIENTRO
2320 FOR I=1 TO 255: LET A=USR 6
1350: NEXT I
2325 GO TO 2400
2330 RESTORE 2340: FOR R=0 TO 24
: READ code: POKE 61350+R,code:
NEXT R: RETURN
2340 DATA 6,64,17,0,72,213,225,3
5,197,1,31,0,26,237,176,43,119,0
,35,35,19,193,16,240,201
2425 PAUSE 200: PAPER 0: CLS : G
O TO 2500
3100 REM creazione di scritte in
rilievo coperte da PAPER 0 per
la creazione di copertine
3120 FOR P=1 TO 8 STEP 1.7: INK
0
3150 PLOT 24+P,136-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-12: DRAW -16
,0: DRAW 16,-12
3155 PLOT 56+P,136-P: DRAW 0,24:
DRAW 16,0: DRAW 0,-24: DRAW -16
,0
3160 PLOT 88+P,160-P: DRAW 0,-24
: DRAW 16,0: DRAW 0,24
3165 PLOT 122+P,160-P: DRAW 26,0
: PLOT 136+P,160-P: DRAW 0,-24
3170 PLOT 166+P,160-P: DRAW 0,-2
4
3175 PLOT 184+P,136-P: DRAW 0,24
: DRAW 16,-24: DRAW 0,24
3180 PLOT 232+P,136-P: DRAW -16,
0: DRAW 0,12: DRAW 10,0: DRAW -1
0,0: DRAW 0,12: DRAW 16,0
3185 NEXT P
3190 PRINT AT 11,2: FLASH 1: BRI
GHT 1:"REALIZZATA DA IVANO PARBU
ONO": FLASH 0
3192 REM segue creazione di scri
tte in rilievo coperte da PAPER
0 per la creazione di copertine
3195 FOR P=1 TO 6 STEP 1.3: INK
0
3200 PLOT 8+P,16-P: DRAW 0,16
3205 PLOT 15+P,32-P: DRAW 4,-16:
DRAW 4,16
3210 PLOT 31+P,16-P: DRAW 4,16:
DRAW 4,-16: PLOT 35+P,24-P: DRAW
4,0
3215 PLOT 48+P,16-P: DRAW 0,16:
DRAW 8,-16: DRAW 0,16
3220 PLOT 64+P,16-P: DRAW 0,16:
DRAW 4,0: DRAW 0,-16: DRAW -4,0
3225 PLOT 128+P,16-P: DRAW 0,16:

```

```

DRAW 8,0: DRAW 0,-8: DRAW -8,0
3230 PLOT 144+P,16-P: DRAW 4,16:
DRAW 4,-16: PLOT 148+P,24-P: DR
AW 4,0
3235 PLOT 150+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-8: DRAW -8,0:
DRAW 8,-8
3240 PLOT 176+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-6: DRAW -3,-2
: DRAW -5,0: DRAW 5,0: DRAW 3,-2
: DRAW 0,-6: DRAW -8,0
3245 PLOT 192+P,32-P: DRAW 0,-16
: DRAW 8,0: DRAW 0,16
3250 PLOT 208+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-16: DRAW -8,0
3255 PLOT 224+P,16-P: DRAW 0,16:
DRAW 8,-16: DRAW 0,16
3260 PLOT 240+P,16-P: DRAW 0,16:
DRAW 8,0: DRAW 0,-16: DRAW -8,0
3265 NEXT P
3290 PRINT #0;"PREMI SPACE PER C
ONTINUARE"
3300 PAUSE 0
3350 REM routine che permette di
scoprire la pagina grafica in p
recedenza realizzata
3450 FOR T=0 TO 175: PLOT 2,174-
T: DRAW INK 7: OVER 1:252,0
3460 NEXT T: INK 7
3500 REM routine che permette lo
scorrimento veloce verso destra
di 1/3 dello schermo
3510 GO SUB 3525
3520 FOR I=1 TO 255: LET A=USR 6
1300: NEXT I
3522 GO TO 3600
3525 RESTORE 3530: FOR R=0 TO 24
: READ code: POKE 61300+R,code:
NEXT R: RETURN
3530 DATA 6,64,17,255,71,213,225
,43,197,1,31,0,26,237,184,35,119
,0,43,43,27,193,16,240,201
3500 REM routine che permette lo
scorrimento veloce verso sinist
ra di 1/3 dello schermo
3510 GO SUB 3530
3520 FOR I=1 TO 255: LET A=USR 6
1400: NEXT I
3525 GO TO 3900
3530 RESTORE 3640: FOR R=0 TO 24
: READ code: POKE 61400+R,code:
NEXT R: RETURN
3640 DATA 6,64,17,0,64,213,225,3
5,197,1,31,0,26,237,176,43,119,0
,35,35,19,193,16,240,201
3900 PAUSE 200: CLS : BRIGHT 1:
PRINT AT 11,13:"F I N E"
3910 PAUSE 0

```

abbiamo chiamato terremoto.

La successiva routine chiamata Creazione di scritte in rilievo varia da quella iniziale perché è stato tolto il salto dando così un effetto di un tutt'uno. I 32 Data della linea 2130 vengono collocati alla locazione di memoria che va dalla 61200 alla 61231; il programma ritorna alla linea 2102, dove il loop For I = 1 To 512 permette lo scroll con rientro negativo e positivo a sinistra a seconda dei valori assegnati nel ciclo e chiaramente si può accorciare o prolungare lo scroll. Terminata questa dimostrazione ne segue subito un'altra realizzata con le stesse caratteristiche, ma variando naturalmente i Data e l'assegnazione degli stessi a differenti locazioni di memoria che per questa routine sono stati assegnati a partire dal 61350 al 61374 (scroll veloce a sinistra con rientro). La durata dello scroll è determinata dalla linea 2320 con il loop For I = 1 To 255. La routine successiva si divide in due parti: la prima crea una pagina grafica a piacere (3100-3280) per mezzo di Plot e Draw. Sono state realizzate delle scritte ma potrebbero essere anche disegni, in pratica tutto ciò che si vuole, l'importante che all'inizio si inserisca l'istruzione Paper 0. Quando il computer ha creato la scritta, in questo caso in rilievo (come spiegato nella prima routine di questa raccolta) e dopo aver premuto la barra spaziatrice entrerà in azione la parte più importante di questa routine, cioè le li-

nee che vanno dalla 3450 alla 3460 in pratica il loop che permette di scoprire il video lasciando intatto ciò che in precedenza era stato realizzato. For T = 0 To 175 stabilisce la quantità di schermo da scoprire, il Plot indica da dove deve iniziare ad essere scoperto il video e il Draw la quantità in orizzontale dello schermo da scoprire. Naturalmente il comando Ink può essere del colore desiderato.

Ma la cosa più importante e forse anche più interessante è quella di provare a cambiare i valori di linea 3450 a proprio piacere e si vedranno delle cose

sempre diverse e molto interessanti che potranno essere magari utili per qualche altra realizzazione (consigliamo proprio, per comprendere in pieno questa routine di fare degli esperimenti). Passiamo ora alle due ultime routine di questa raccolta: la prima permette lo "scorrimento veloce verso destra di 1/3 dello schermo" (3510-3530). Essa, in pratica, agisce come tutte le altre precedentemente descritte; la linea 3520, per mezzo del ciclo For I = 1 To 255, stabilisce la quantità di scorrimento che si desidera per questa routine.

L'ultima permette lo "scorrimento velo-

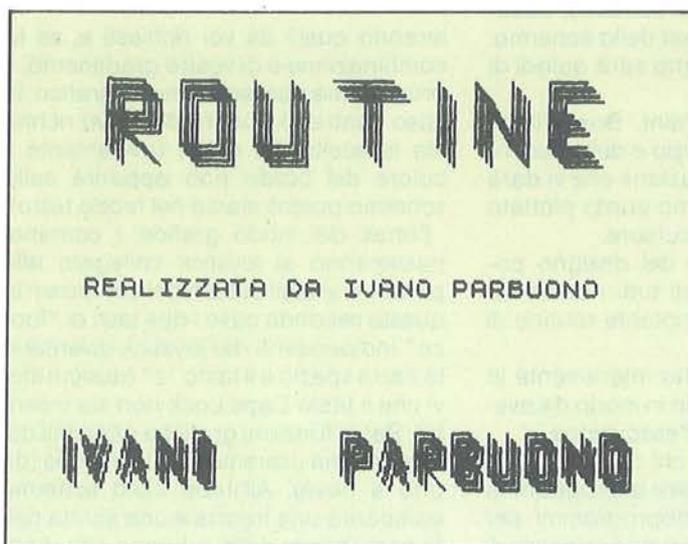


Figura 1 - Rappresentazione in rilievo di scritte.

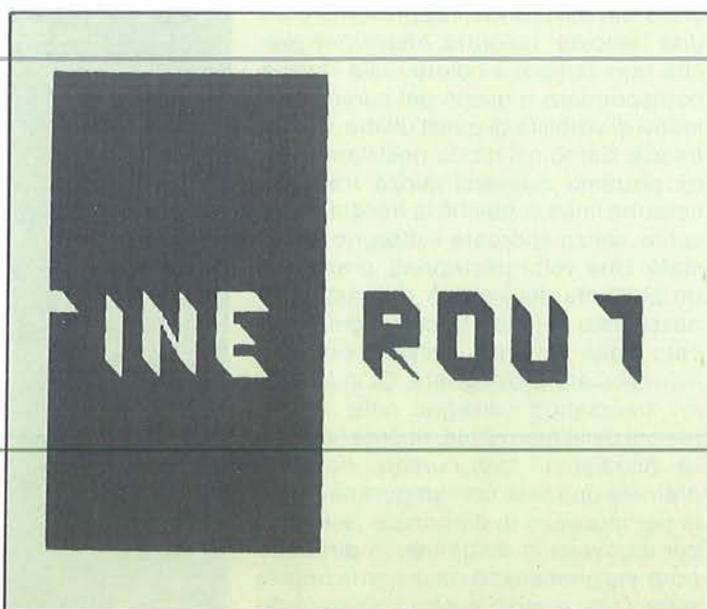


Figura 2 - Rappresentazione grafica di scroll laterale con rientro negativo.

ce verso sinistra di 1/3 dello schermo": qui variano solamente alcuni Data di linea 3640 e il numero di partenza della loro assegnazione in memoria, che in questo caso vanno dalla locazione 61400 alla 61424.

Proprio in relazione all'assegnazione dei Data in memoria si deve dire che non è assolutamente tassativo assegnarli così come li abbiamo collocati, ma si possono dare delle locazioni diverse: ad esempio, invece di 61400 si potrebbe partire benissimo dal 31000 oppure dal 62500, cioè in base alla disponibilità che si ha nel programma nel quale si vuole inserire la routine che ci interessa. ■

**Q**uesto programma vi permetterà di disegnare in alta risoluzione mediante il joystick o, per chi ne fosse sprovvisto, con i 4 tasti cursore sfruttando tutte le istruzioni grafiche del BASIC MSX, richiamabili mediante i tasti funzione del computer.

Il programma lungo 8 Kbyte, fa un ampio uso degli sprite per poter visualizzare i vari cursori utilizzati nel programma senza pericolo di "sporcare" il disegno sottostante. Durante l'esecuzione del programma numerosi messaggi vi guideranno nelle varie fasi della stesura del disegno. Questi messaggi appariranno nella parte bassa dello schermo, occupando gli ultimi 10 pixel dello schermo: l'area utile per il disegno sarà quindi di 181 x 255 pixel.

Oltre alle funzioni Paint, Box, Ellisse (con controllo del raggio e della eccentricità), troverete l'istruzione che vi darà le coordinate dell'ultimo punto plottato o della posizione del cursore.

Durante l'esecuzione del disegno potranno essere utilizzati tutti i colori disponibili oltre a una potente routine di cancellazione.

Il programma è scritto interamente in BASIC, ma è strutturato in modo da avere una alta velocità di esecuzione.

Questo consentirà a chi ne sentisse la necessità, di aggiungere al programma delle istruzioni o sottoprogrammi per renderlo più adatto alle varie esigenze di utilizzo.

Anche in questa configurazione, il programma risulta comunque sufficientemente completo e pratico, soprattutto se utilizzerete un joystick con due pulsanti di "fuoco" indipendenti, come prescritto dallo standard MSX (il joystick andrà collegato alla porta 1 del computer).

Se volete mettere alla prova la vostra fantasia e capacità nella "computer art", non vi resta che cominciare a digitare il (lungo) programma: la fatica vi sarà senz'altro ricompensata con ore di divertimento!

### Descrizione del programma

Dopo la schermata introduttiva di presentazione, vi verrà richiesto il colore dello sfondo della prima traccia e del bordo del piano del disegno. Effettuata la scelta, i colori della schermata diven-

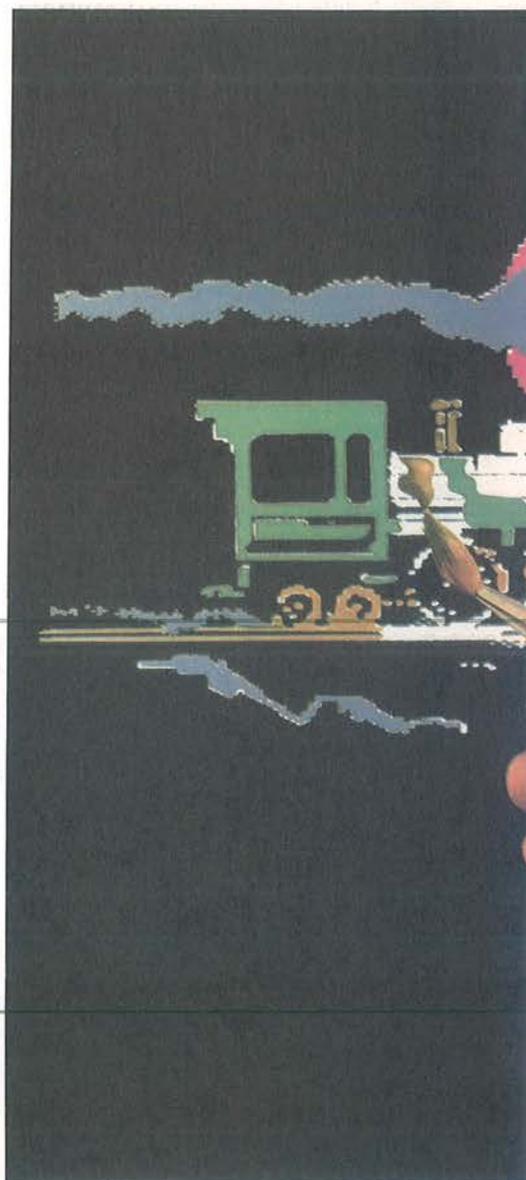
# Disegnatore per MSX

Date libero sfogo alla vostra fantasia

di Paolo Ferrami

teranno quelli da voi richiesti e, se la combinazione è di vostro gradimento, il programma passerà al modo grafico; in caso contrario vi verrà di nuovo richiesta la scelta dei colori (ovviamente il colore del bordo non apparirà sullo schermo poiché siamo nel modo testo).

Entrati nel modo grafico, i comandi passeranno al joystick collegato alla porta 1 o ai tasti cursori del computer: in questo secondo caso i due tasti di "fuoco" indipendenti dal joystick diventano la barra spazio e il tasto "z" (assicuratevi che il tasto Caps Lock non sia inserito). Per le funzioni grafiche ottenibili dal programma useremo i tasti funzione (da uno a nove). All'inizio sullo schermo comparirà una freccia e una scritta nella parte bassa dello schermo che vi comunicherà lo stato del sistema e il colore che si sta utilizzando, rappresentato da una "faccina" colorata. Attenzione perché non sempre il colore della traccia corrisponderà a quello del cursore per motivi di visibilità di quest'ultimo. Con la freccia siamo nel modo posizionamento: potremo muoverci senza tracciare nessuna linea e, poiché la freccia è uno sprite, senza sporcare il disegno sottostante. Una volta posizionati, premendo un pulsante del joystick e il tasto "z", passeremo al modo traccia, come indicato dalla scrittura bassa e potremo incominciare a disegnare. Ci muoveremo, tracciando il disegno, nelle otto direzioni della manopola, ricordando che se utilizziamo i tasti cursore, dovremo premere due tasti contemporaneamente per muoverci in diagonale (esempio: per muoverci in diagonale in direzione nord-est premeremo i tasti con la freccia verso l'alto e verso destra contempora-



neamente). Per ritornare al modo posizionamento premere l'altro pulsante di "fuoco" del joystick o la barra spazio. Per il disegno abbiamo a disposizione nove utility, ciascuna richiamabile mediante i tasti funzione: vediamo a cosa servono e come funzionano.

**Tasto F1** - Traccia una linea con inclinazione a piacere. Questo comando è risultato necessario perché con il joystick possiamo tracciare linee con inclinazione fissa di 45°. Una volta premuto questo tasto, comparirà sullo schermo uno sprite rettangolare con un pixel acceso internamente. Spostiamoci nella posi-

zione in cui vogliamo avere il secondo estremo del segmento, utilizzando come riferimento il punto interno al rettangolo.

Premendo il pulsante del joystick o la barra spaziatrice verrà tracciato il segmento. È possibile, volendo, eseguire tutto il disegno utilizzando questa funzione, senza tracciare nessuna linea in modo diretto: l'unico limite sarà posto dalla difficoltà di tracciare linee curve (che in questo programma dovranno essere approssimate a delle spezzate poligonali).

**Tasto F2** - Scelta dei colori da 1 a 9. Premendo F2 comparirà sulla riga dedicata ai messaggi la serie di colori che vanno da 1 a 9: sarà sufficiente premere il tasto corrispondente al numero del colore scelto per cambiarlo.

**Tasto F3** - Colori da 10 a 15: premere il tasto corrispondente all'unità che caratterizza il colore che si vuole utilizzare (esempio: colore 15, premi il tasto 5).

**Tasto F4** - Cancellazione. Questo comando ha una duplice funzione; quella di cancellare delle linee sullo schermo come una normale gomma e quella di disegnare con il colore del fondo su aree di altro colore (per esempio riempite con l'istruzione Paint). Sullo schermo comparirà uno sprite a forma di croce con all'interno il pixel di riferimento. Questa croce possiamo posizionarla in qualunque punto del video senza pericolo di cancellare il disegno sottostante: la cancellazione avverrà infatti se contemporaneamente al movimento del joystick terremo premuto un pulsante di "fuoco" o la barra spaziatrice. Con l'altro pulsante della manopola o col tasto "z" torneremo al modo posizionamento.

**Tasto F5** - Disegno di un cerchio o di una ellisse. Appena premuto questo tasto comparirà sullo schermo un piccolo cerchio che avrà come centro l'ultima posizione assunta dal cursore, e una croce simile a quella usata per cancellare. Muovendo il joystick ci posizioneremo nel punto in cui dovrà cadere l'estremo del semiasse; fatta questa operazione premiamo un pulsante della manopola o lo spazio e ci verrà richiesta l'eccentricità: premendo 1 il computer tratterà un cerchio con un numero compreso tra 2 e 5 avremo ellissi con asse orizzontale maggiore dell'asse verticale, il contrario per numeri compresi tra 6 e 0.

Tracciata la curva torneremo al modo posizionamento. Per motivi di praticità, il range dei valori che danno l'eccentricità è limitato: chi volesse può modificare i valori nel listato per cambiare o ampliare il range.

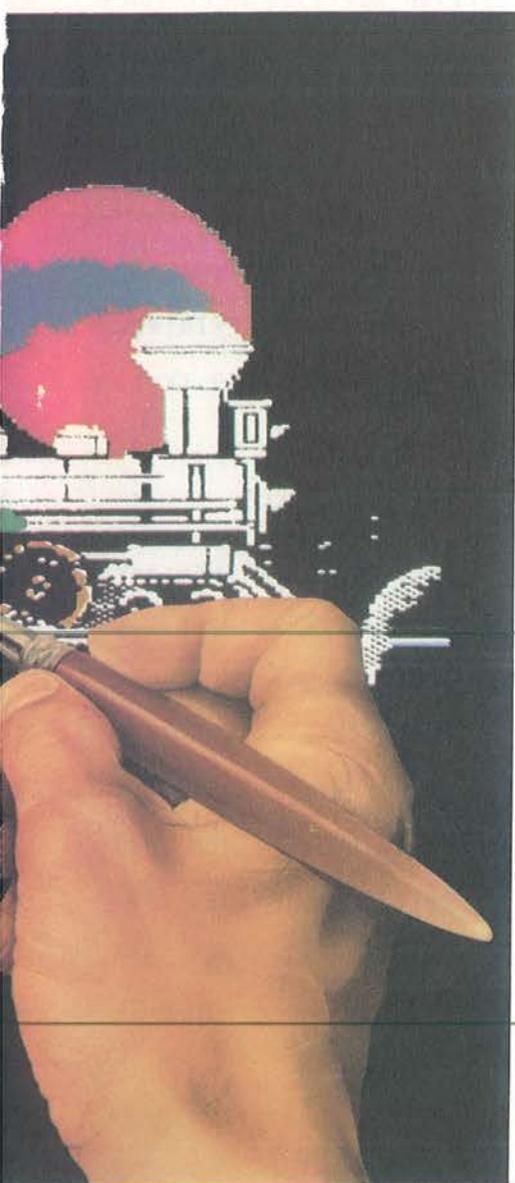
**Tasto F6** - Dà le coordinate del cursore. Dopo aver premuto questo tasto apparirà sulla riga riservata ai messaggi l'ascissa e l'ordinata del cursore, sia che ci si trovi nel modo posizionamento che nel modo traccia. È utile usare questa funzione prima di tracciare un cerchio in modo da conoscere esattamente le coordinate del centro.

**Tasto F7** - Disegna rettangoli. Sullo schermo apparirà lo sprite già utilizzato per tracciare le linee (F1) oltre a un pixel indicante la posizione di uno dei vertici del rettangolo. Utilizzando la manopola (o i cursori), sposteremo il cursore nel punto in cui desideriamo l'altro estremo della diagonale del rettangolo: premendo uno dei pulsanti del joystick o la barra spaziatrice, verrà visualizzato il rettangolo. Anche qui, come in tutti gli altri casi, le segnalazioni che appariranno sulla parte bassa del video aiuteranno a eseguire correttamente le operazioni.

**Testo F8** - Riempimento di una superficie (Paint). Questa funzione attiva l'istruzione Paint del BASIC MSX.

Vediamo come si deve procedere per utilizzarla correttamente: mettiamoci nel modo posizionamento e portiamo la freccia all'interno della figura che vogliamo colorare; controlliamo che i confini della figura siano perfettamente chiusi: anche un'apertura di un solo pixel potrebbe causare il riempimento di tutto lo schermo, cancellando in tale modo tutto il disegno. Poiché l'istruzione Paint riconosce come frontiera solo i perimetri dello stesso colore di quello utilizzato per colorare, controllate che il colore della frontiera della figura sia lo stesso della "faccina" sulla riga dei messaggi. Fatti questi controlli, potete procedere al riempimento.

**Testo F9** - Fine disegno. Questa istruzione cancella la riga dei messaggi e toglie gli sprite, bloccando il programma nel modo Screen 2 (modo grafico ad alta risoluzione). Per uscire (cancellando il disegno) premete i tasti Ctrl e Stop. Il tasto F10 non è stato utilizzato per fare in modo che possa essere riservato a particolari routine (testo, salvataggio su



Listato 1 - Il programma Disegnatore per MSX.

```

1 REM *****
2 REM
3 REM   DISEGNATORE - M.S.X.
4 REM
5 REM       di Paolo Ferrami
6 REM
7 REM       1985
8 REM
9 REM *****
10 REM
20 REM PRESENTAZIONE
30 REM
35 WIDTH(39)
40 SCREEN 2:COLOR9,1,1:CLS
50 OPEN"GRP:"AS1
60 DEFINT U
70 CIRCLE(127,100),90,9,,.6:BEEP
80 CIRCLE(127,100),70,9,,.6:BEEP
90 PAINT(50,100),9:BEEP
100 CIRCLE(127,100),30,4,,.6:BEEP
110 CIRCLE(127,100),20,4,,.6:BEEP
120 PAINT(100,100),4:BEEP
130 LINE(0,0)-(60,16),15,B
160 PSET(82,30),1:BEEP
170 FOR A =1 TO 11:READ A$
180 U=RND(1)*15:IFU=1THEN U=2
190 COLORU:PRINT#1,A$;:NEXTA
200 PSET(118,97),1
210 COLOR15:PRINT#1,"MSX"
220 PSET(65,167),1
230 COLOR10:PRINT#1,"1985 by P.Ferrami":
CLOSE#1:GOSUB2490
240 DATA D,I,S,E,G,N,A,T,O,R,E
250 REM
260 REM     MENU
265 REM
270 O=3:X=100:Y=100:I=15:IN=1
280 KEY OFF
290 COLOR12,1:SCREEN0
300 L$="*****
*****"
310 PRINTL$:LOCATE9,3:PRINT"DISEGNATORE
PER M.S.X."
320 PRINT:PRINTL$:PRINT
330 INPUT"COLORE DELLA CARTA (1-15)";CA
340 PRINT:INPUT"COLORE DELLA PRIMA TRACC
IA (1-15)";I
350 PRINT:INPUT"COLORE DEL BORDO
(1-15)";B0
360 PRINT:PRINT:PRINTL$
370 COLORI,CA,CA
380 INPUT"CONFERMI? (S/N)";S$
390 IFS$="N"OR S$="n" THEN 290
400 REM

```

```

410 REM   DISEGNO
420 REM
430 SCREEN2,0:COLORI,CA,B0:CLS
440 OPEN"GRP:"AS1
450 IN=I
460 IFCA=1THENQ=4ELSEQ=1
470 LINE(0,183)-STEP(255,9),Q,BF
480 REM
490 REM ALLA ROUTINE SPRITE
500 REM
510 GOSUB2100
520 REM
530 REM ALLA ROUTINE CONTROLLO CURSORI
540 REM
550 GOSUB 620:KEY(1)ON:KEY(2)ON:KEY(3)ON
:KEY(4)ON:KEY(5)ON:KEY(6)ON:KEY(7)ON:KEY
(8)ON:KEY(9)ON
560 IF O=1 THEN :PRESET(3,183):COLORQ:PR
INT#1,"*****"
:COLOR15:PRESET(8,183):PRINT#1,"M
ODO: TRACCIA ";:COLORI:PRINT#1,"●":O
=11:COLORI
570 IFO=10RO=11THENPSET(X,Y)
580 IF O=3 THEN :PRESET(3,183):COLORQ:PR
INT#1,"*****"
:COLOR15:PRESET(8,183):PRINT#1,"M
ODO: POSIZIONAMENTO ";:COLORI:PRINT#1,
"●":O=10:COLOR I
590 IF O=30RO=10 THEN PUT SPRITE 9,(X-8,
Y-4),IN,0
600 GOTO 550
610 REM
620 REM CONTROLLO CURSORI
630 REM E JOYSTICK
640 REM
650 IF STICK(0)=1 OR STICK(1)=1 THEN Y=Y
-1
660 IF STICK(0)=2 OR STICK(1)=2THENX=X+1
:Y=Y-1
670 IF STICK(0)=3 OR STICK(1)=3THEN X=X+
1
680 IF STICK(0)=4 OR STICK(1)=4THEN X=X+
1:Y=Y+1
690 IF STICK(0)=5 OR STICK(1)=5THEN Y=Y+
1
700 IF STICK(0)=6 OR STICK(1)=6THEN Y=Y+
1:X=X-1
710 IF STICK(0)=7 OR STICK(1)=7THEN X=X-
1
720 IF STICK(0)=8 OR STICK(1)=8THEN X=X-
1:Y=Y-1
730 IFY>182 THEN Y=182:BEEP
740 IF Y<0 THEN Y=0:BEEP
750 IFX>254 THEN X=254 :BEEP
760 IFX<0 THEN X=0:BEEP
770 IF STRIG(1)=-1 OR INKEY$="z"OR INKEY

```

disco/cassetta ecc.) non comprese in questo programma, ma facilmente inseribili nel listato.

## REMARKS

**1-250** - Presentazione del programma. Nella riga 50 viene aperto un file per poter scrivere, mediante l'istruzione Print# 1 nella pagina grafica. Per posizionare le scritte viene usata l'istruzione Preset. La linea 230 chiude il file GRP.

**260-390** - Scelta del colore e verifica della compatibilità.

**400-470** - Inizio disegno. Nella linea 440

viene riaperto il file GRP. Nella linea 460 viene controllato che il colore degli sprite sia diverso da quello del fondo per evitare che risultino invisibili. Nella linea 470 viene colorato il fondo della riga dei messaggi.

**480-520** - Gosub alla routine di caricamento degli sprite.

**530-540** - Gosub alla routine di controllo del movimento del joystick e dei tasti cursori.

**550-600** - Nella linea 550 vengono attivati 9 tasti-funzione; nelle seguenti linee si controlla il valore della variabile-flag O per verificare se siamo in modo posi-

zionamento o in modo traccia.

**600-800** - Subroutine controllo cursori-joystick. Nelle linee 660-720 si controllano le risposte del joystick inserito nella porta 1 e dei tasti cursore. Le 4 linee da 730 a 760 controllano che la traccia non esca dai confini del video: nel caso si raggiunga il bordo verrà fermato l'incremento delle coordinate e un segnale sonoro avvertirà l'utilizzatore della fine dello spazio disponibile per la traccia. Le linee 770 e 780 controllano lo stato dei pulsanti del joystick oltre che la barra spaziatrice e del tasto (Z). La linea 790, infine, indirizza alle subroutine ri-

### Seguito listato Disegnatore per MSX.

```

$="Z" THEN O=3
780 IF STRIG(0)=-1 OR STRIG(3)=-1 THEN O
=1:PUT SPRITE9,(X-8,Y-4),0,0
790 ON KEY GOSUB810,980,1090,1160,1280,1
730,1800,1900,2000
800 RETURN
810 REM
820 REM OPZIONE LINEA
830 REM
840 BEEP
850 PSET(X,Y):PUTSPRITE9,(X+8,Y+4),0,0:
PRESET(3,183):COLORQ:PRINT#1,"
":COLOR15:F
ORA=1T05:PRESET(8,183):PRINT#1,"LINEA: T
RACCIA L'ALTRO ESTREMO":COLORI
860 QW=X:ER=Y
870 GOSUB610
880 PUT SPRITE 5,(X-4,Y-4),IN,5
890 IF STRIG(0)=-1 OR STRIG(3)=-1 THEN O
=1:PUT SPRITE5,(X-4,Y-4),IN,5:GOTO910
900 GOTO870
910 LINE(QW,ER)-(X,Y),I:PUTSPRITE5,(X-4,
Y-4),0,5
920 PRESET(3,183):COLORQ:PRINT#1,"
":COLORI
930 RETURN
940 O=10:RETURN
950 REM
960 REM COLORE DA 1 A 9
970 REM
980 PRESET(3,183):COLORQ:PRINT#1,"
":COLORI
990 IFQ=4THENPRESET(3,183):COLOR15:PRINT
#1,"COLORE 1->9 ";:FORA=1T03:COLORA:PRIN
T#1,"● ";:NEXT:PRINT#1,"@ ";:FORA=5T08:C
OLORA:PRINT#1,"● ";:NEXT:COLOR9:PRINT#1,
"● ":COLORI

```

```

1000 IFQ=1THENPRESET(3,183):COLOR15:PRIN
T#1,"COLORE 1->9 @ ";:FORA=2T08:COLORA:P
RINT#1,"● ";:NEXT:COLOR9:PRINT#1,"● ":CO
LORI
1010 BEEP:I$=INKEY$:IFI$=" THEN1010
1020 I=ASC(I$):I=I-48:IN=I
1030 IFI<=0 ORI>=10THEN GOTO 980
1040 PRESET(3,183):COLORQ:PRINT#1,"
":COLORI:O
=3
1050 RETURN
1060 REM
1070 REM COLORE DA 10 A 15
1080 REM
1090 PRESET(3,183):COLORQ:PRINT#1,"
":COLORIN
1100 BEEP:PRESET(8,183):COLOR15:PRINT#1
,"COLORE 1->5 (+10) ";:FORA=11T014:COLOR
A:PRINT#1,"● ";:NEXT:COLOR15:PRINT#1,"●
":COLORIN
1110 BEEP:I$=INKEY$:IFI$=" THEN1110
1120 I=ASC(I$):I=I-38:IN=I
1130 IFI<=10 ORI>=16THEN GOTO 1110
1140 PRESET(3,183):COLORQ:PRINT#1,"
":COLORI
1150 O=3:RETURN
1160 REM
1170 REM CANCELLAZIONE
1180 REM
1190 PUT SPRITE9,(X-4,Y-4),0,0
1200 PRESET(3,183):COLORQ:PRINT#1,"
":COLORI
1210 PRESET(3,183):COLOR15:PRINT#1,"CAN
CELLAZIONE ->";:COLOR10:PRINT#1,"[FIRE]"
:COLORI
1220 PUT SPRITE7,(X-3,Y-4),IN,3
1230 IFSTRIG(1)=-1 OR STRIG(0)=-1 THEN P
RESET(X,Y),CA
1240 GOSUB620

```

# OLTRE LE BARRIERE D

RAM

80 CD  
RS-  
DISK

MODEM

FILE

SVI  
E

GAME

# SV

## SPECTRUM

il computer del gra

Distributore per l'Italia COMTRAD Divisione Cor

# ELL'HARD E DEL SOFT.

LUMIN  
232  
DRIVE

PRINTER

232

728

232

**M**<sup>TM</sup>

**AVIDEO**

**nde standard MSX**

puters Tel. (0586) 424348 TLX 623481 COMTRD



Seguito listato Disegnatore per MSX.

```

1250 IF STRIG(3)=-10R INKEY$="z"OR INKEY
$="Z"THEN PUT SPRITE7,(X-3,Y-4),0,3:PRES
ET(3,183): COLORQ:PRINT#1,"
":COLORIN:RETU
RN
1260 GOTO1220
1270 REM
1280 REM CERCHI
1290 REM
1300 A=X
1310 BEEP
1320 PUT SPRITE 9,(X-4,Y-4),0,0
1330 PRESET(3,183): COLORQ:PRINT#1,"
":COL
ORIN
1340 PRESET(3,183):COLOR15:PRINT#1,"CER
CHIO-ELLISSE -> RAGGIO":COLORIN
1350 PUT SPRITE1,(A-4,Y-4),IN,4
1360 IFSTICK(1)=3 OR STICK(0)=3 THEN X=X
+1
1370 IFSTICK(1)=7 OR STICK(0)=7 THEN X=X
-1
1380 PUT SPRITE2,(X-4,Y-4),IN,3
1390 P=0:IF STRIG(1)=-1 OR STRIG(0)=-1 T
HEN PRESET(3,183):COLORQ:PRINT#1,"
":GOTO1
420
1400 GOTO1350
1410 REM
1420 REM CERCHIO O ELLISSE
1430 REM CONTROLLO ECCENTRICITA'
1440 REM
1450 PRESET(3,183):COLOR15:PRINT#1,"ECC
ENTRICITA' (1=CERCHIO)":COLORI
1460 LETI$=INKEY$
1470 IFI$="1"THEN BEEP:E=1.4:GOTO1590
1480 IFI$="2"THEN BEEP:E=2:GOTO1590
1490 IFI$="3"THEN BEEP:E=2.5:GOTO1590
1500 IFI$="4"THEN BEEP:E=3:GOTO1590
1510 IFI$="5"THEN BEEP:E=3.5:GOTO1590
1520 IFI$="6"THEN BEEP:E=.1:GOTO 1590
1530 IFI$="7"THEN BEEP:E=.3:GOTO 1590
1540 IFI$="8"THEN BEEP:E=.5:GOTO 1590
1550 IFI$="9"THEN BEEP:E=.7:GOTO 1590
1560 IFI$="0"THEN BEEP:E=.9:GOTO1590
1570 BEEP:GOTO1460
1580 REM
1590 REM DISEGNO CERCHIO-ELLISSE
1600 REM
1610 PRESET(3,183):COLORQ:PRINT#1,"
":COLO
RIN
1620 P=0:BEEP:PUT SPRITE 3,(8,19),P,10:P
UTSPRITE4,(16,19),P,11:PUTSPRITE5,(24,19

```

```

),P,12
1630 LETB=ABS(X-A)*E
1640 PUT SPRITE 7,(100,170),0,4
1650 PUT SPRITE1,(A,B),0,4
1660 PUT SPRITE2,(X,Y),0,3
1670 IF E>1 THEN CIRCLE(A,Y),B,I,,,E: O=
3
1680 IF E<1 THEN CIRCLE(A,Y),B/E,I,,,E:
O=3
1690 GOTO 590
1700 REM
1710 REM COORDINATE
1720 REM
1730 PRESET(3,183):COLORQ:PRINT#1,"
":COLO
R15:FORA=1TO5:PRESET(8,183):PRINT#1," X
=",X;" Y=";Y:NEXTA:BEEP
1740 IF O=1 OR O=11 THEN PRESET(3,183):C
OLORQ:PRINT#1,"
":COLOR15:PRESET(8,183):PR
INT#1,"MOD0: TRACCIA ";:COLORI:PRINT
#1,"0=11:COLORI
1750 IF O=3OR O=10 THEN PRESET(3,183):C
OLORQ:PRINT#1,"
":COLOR15:PRESET(8,183):PR
INT#1,"MOD0: POSIZIONAMENTO ";:COLORI:
PRINT#1,"0=10:COLOR I
1760 RETURN
1770 REM
1780 REM RETTANGOLI
1790 REM
1800 PSET(X,Y):PUTSPRITE9,(X+8,Y+4),0,0:
PRESET(3,183):COLORQ:PRINT#1,"
":COLOR15:
FORA=1TO5:PRESET(8,183):PRINT#1,"RETTANG
OLO:TRACCIA LA DIAGONALE":COLORI
1810 QW=X:ER=Y
1820 GOSUB650
1830 PUT SPRITE 5,(X-4,Y-4),IN,5
1840 IF STRIG(0)=-1 OR STRIG(3)=-1 THEN
O=1:PUT SPRITE5,(X-4,Y-4),IN,5:GOTO1860
1850 GOTO1820
1860 BEEP:LINE(QW,ER)-(X,Y),I,B:PUT SPRI
TE5,(X-4,Y-4),0,5
1870 IF O=1 OR O=11 THEN PRESET(3,183):C
OLORQ:PRINT#1,"
":COLOR15:PRESET(8,183):PR
INT#1,"MOD0: TRACCIA ";:COLORI:PRINT
#1,"0=11:COLORI
1880 IF O=3OR O=10 THEN PRESET(3,183):C
OLORQ:PRINT#1,"
":COLOR15:PRESET(8,183):PR
INT#1,"MOD0: POSIZIONAMENTO ";:COLORI:
PRINT#1,"0=10:COLOR I
1890 RETURN
1900 REM

```

## Seguito listato Disegnatore per MSX.

```

1910 REM PAINT
1920 REM
1930 BEEP
1940 PAINT(X,Y),I
1950 IFIN=IAND I<13 THEN LETIN=I+3
1960 IF IN =IAND I>12 AND NOT CA=1 THEN
LET IN=1 ELSE IN=10
1970 PRESET(3,183):COLORQ:PRINT#1,"
";COLORI
1980 COLOR15:PRESET(8,183):PRINT#1,"MODO
: POSIZIONAMENTO ";:COLORI:PRINT#1,"●"
:O=10:COLOR I
1990 O=10:RETURN
2000 REM
2010 REM FINE PROGRAMMA
2020 REM
2030 PRESET(3,183):COLORB0:PRINT#1,"
";:LIN
E(0,191)-STEP(255,0):COLORI
2040 PUTSPRITE9,(X-8,Y-4),0,0
2050 GOSUB2490
2060 GOTO2060
2070 REM
2080 REM CARICAMENTO SPIRTE
2090 REM
2100 REM
2110 REM
2120 REM SPRITE FRECCIA
2130 REM
2140 B$="":FOR U=1 TO 8
2150 READ A$
2160 B$=B$+CHR$(VAL("&B"+A$))
2170 NEXT U
2180 SPRITE$(0)=B$
2190 DATA 00001000,00001100,00001110,111
11111,11111111,00001110,00001100,0000100
0
2200 REM SPRITE CROCE
2210 REM
2220 B$="":FOR U=1 TO 8
2230 READ A$
2240 B$=B$+CHR$(VAL("&B"+A$))
2250 NEXT U
2260 SPRITE$(3)=B$
2270 DATA 00010000,00010000,00010000,111
01111,00010000,00010000,00010000,0001000
0
2280 REM
2290 REM SPRITE CERCHIO
2300 REM
2310 B$="":FOR U=1 TO 8
2320 READ A$
2330 B$=B$+CHR$(VAL("&B"+A$))
2340 NEXT U

```

```

2350 SPRITE$(4)=B$
2360 DATA 00011000,00111100,00111100,011
11110,01111110,00111100,00111100,0001100
0
2370 REM
2380 REM SPRITE RETTANGOLO
2390 REM
2400 B$="":FOR U=1 TO 8
2410 READ A$
2420 B$=B$+CHR$(VAL("&B"+A$))
2430 NEXT U
2440 SPRITE$(5)=B$
2450 DATA 00000000,01111110,01000010,010
10010,01000010,01000010,01111110,0
2460 RETURN
2470 REM
2480 REM MUSICA:
2490 REM
2500 REM MINUETTO IN SOL MAGGIORE
2510 REM J.S.BACH
2520 REM
2530 PLAY"T18005L4D04L86AB05C","T18003L2
6L46","T18003L2BR4"
2540 PLAY"05L4D0466","L2B."
2550 PLAY"05L4EL8CDEF+","04L2C."
2560 PLAY"05L4G04L466","03L2B."
2570 PLAY"05L4CL8DC04BA","03L2A."
2580 PLAY"04L4B05L8C04L8BA6","03L26."
2590 PLAY"04L4F+L86AB6","04L4D03L4B6"
2600 PLAY"04L4BL2A","04L4B03L8D04L8C03L8
BA"
2610 PLAY"05L4D04L86AB05C","03L26L46","0
3L2BR4"
2620 PLAY"05L4D0466","03L46B6"
2630 PLAY"05L4EL8CDEF+","04L2C."
2640 PLAY"05L4G04L466","03L4B04L8C03L8B6
A"
2650 PLAY"05L4CL8DC04BA","03L2A14F+"
2660 PLAY"04L4B05L8C04L8BA6","03L26L4B"
2670 PLAY"04L4AL8BA6F+","04L4CD03L4B"
2680 PLAY"04L26.","03L26.","03L2B."
2690 RETURN

```

# Novità Jackson.

G. Loveday - C. Miere

## MISURE ELETTRONICHE E DIAGNOSI DEI GUASTI

Un libro che completa l'istruzione del tecnico elettronico nel campo della rilevazione e delle eliminazione dei guasti dei circuiti elettronici.

Cod. 618P Pag. 368 Lire 34.500

Roberto Doretti

## DATA BASE concetti e disegno

Il passato e il presente di un software applicativo fondamentale nell'EDP, per l'utente, il programmatore e il responsabile di archivi elettronici.

Cod. 526P Pag.186 Lire 22.500

B.W. Kernighan, D.M. Ritchie

## LINGUAGGIO C

Il testo "ufficiale" sul linguaggio C, scritto da chi l'ha progettato e implementato.

Cod. 541P pag. 216 Lire 21.000

Matteuzzi, Pelizzardi

## AMBIENTE UNIX

Un libro che unisce le caratteristiche di un trattato teorico e di un manuale di riferimento, con una completa trattazione del linguaggio C.

Cod. 543P Pag. 198 Lire 19.000

R. Farabone - R. Viano

## LOGICA E DIAGRAMMI A BLOCCHI

Il primo libro che affronta in modo sistematico ed esauriente i diagrammi a blocchi, passaggio obbligato per chiunque debba programmare un elaboratore.

Cod. 539A Pag. 392 Lire 37.000

ritagliare (o fotocopiare) e spedire in busta chiusa a:  
**GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano**

**CEDOLA DI COMMISSIONE LIBRARIA**

**VOGLIATE SPEDIRMI**

n° copie	codice	Titolo	Prezzo unitario	Prezzo totale
			Totale	

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esecuzione del contributo spese di spedizione:

Allego assegno della Banca       Allego fotocopia del versamento sul c/c n. 11666203 a voi intestato

N° \_\_\_\_\_  Allego fotocopia di versamento su vaglia postale e voi intestato

Nome e Cognome \_\_\_\_\_  
 Via \_\_\_\_\_  
 Cap \_\_\_\_\_ Città \_\_\_\_\_ Prov. \_\_\_\_\_  
 Data \_\_\_\_\_ Firma \_\_\_\_\_

Spazio riservato alle Aziende Si richiede l'emissione di fattura      **ORDINE MINIMO L. 50.000**

Partita I.V.A. \_\_\_\_\_

Tullio Trincardi

## LA CONTABILITÀ COL PERSONAL COMPUTER

Diventare protagonisti diretti nell'utilizzo del computer per le proprie esigenze. È l'obiettivo di un test che affronta il problema dai termini generali fino alle specifiche routine di utilità per la costruzione di programmi gestionali complessi e all'esame di alcuni dei pacchetti più utili alle aziende.

Cod. 570P Pag. 280 Lire 27.000



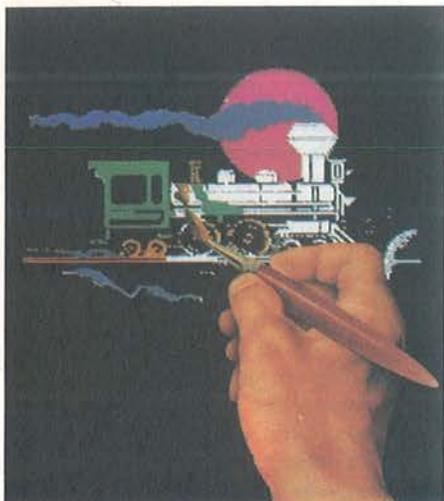
**GRUPPO EDITORIALE JACKSON**

**La biblioteca che fa testo**

chiamate dai tasti funzione.

**810-940** - Subroutine linee inclinate. Dopo aver segnalato l'opzione nella zona dei messaggi, accende gli sprite dedicati (linea 850). Controllati gli impulsi dal joystick, sposta lo sprite alle nuove coordinate e, se viene premuta la barra spaziatrice o un pulsante del joystick, traccia la linea, uscendo così dal sottoprogramma.

**950-1050** - Subroutine colori. Mostrati i



colori disponibili, il programma si predispone ad accettare dalla tastiera il numero del colore prescelto (1010-1020); controllata la validità del numero impostato, torna alla routine principale. Le linee 990 e 1000 verranno utilizzate a seconda del colore della linea dei messaggi in modo che il simbolo con lo stesso colore della linea venga disegnato solo col contorno.

**1060-1150** - Subroutine colori da 10 a 15.

**1160-1260** - Subroutine cancellazione. Segnalata l'opzione, va alla subroutine di controllo movimento. Se la barra spaziatrice o un pulsante del joystick è premuto, il programma effettua la cancellazione mediante l'istruzione Preset (linea 1230). Premendo l'altro pulsante ritornerà alla routine principale.

**1270-1690** - Subroutine cerchio-ellisse. Questo sottoprogramma è strutturato in tre blocchi: il primo (1270-1400) prepara lo schermo e fa apparire gli sprite necessari alla scelta del raggio (o all'asse orizzontale). Le linee 1360-1370 controllano il movimento del joystick mentre la linea 1390 manderà al secondo blocco,

dedicato alla scelta dell'eccentricità (1410-1570). Qui, a seconda del tasto premuto, avremo un diverso valore di eccentricità. Il terzo blocco, dopo aver elaborato il valore dell'eccentricità in modo che la curva abbia effettivamente il raggio richiesto (linea 1630), effettuerà il disegno.

**1700-1760** - Subroutine coordinate. Stampa sulla linea dei messaggi le coordinate X e Y dell'ultimo punto plottato per un tempo determinato del ciclo For-Next della linea 1730.

**1770-1890** - Subroutine rettangoli. La struttura della subroutine è molto simile a quella utilizzata per tracciare le linee: viene utilizzata ancora l'istruzione Line del BASIC MSX, con la sola aggiunta della lettera B che tratterà il rettangolo di cui Line è la diagonale (1860).

**1900-1990** - Subroutine Paint. Questa routine sfrutta semplicemente l'istruzione MSX Paint. Le linee 1950 e 1960 sono due istruzioni di controllo If-Then-Else che danno allo sprite-cursore un colore diverso da quello del fondo e della traccia, per far sì che lo sprite sia sempre visibile e non si confonda col disegno.

**2000-2060** - Fine programma. Toglie tutti gli sprite e i messaggi del video e, dopo

aver lanciato la routine musicale, blocca il programma alla linea 2060 per evitare che ritorni nel modo testo, provocando la cancellazione del disegno.

**2070-2460** - Subroutine caricamento degli sprite. È formata da quattro blocchi, ciascuno dei quali carica uno sprite leggendo i numeri binari nei Data.

**2470-2690** - Subroutine musica. È un minuetto in Sol maggiore di J.S.Bach, suonato utilizzando i tre canali dell'istruzione Play. ■

#### Variabili utilizzate

- A** - Variabile di controllo e ascissa centro del cerchio.
- U** - Variabile di controllo.
- D** - Flag di controllo del modo grafico.
- X** - Ascissa dell'ultimo punto disegnato.
- Y** - Ordinata dell'ultimo punto disegnato.
- I** - Colore traccia e codice ASCII di I\$.
- CA** - Colore fondo.
- IN** - Colore sprite.
- Q** - Colore linea messaggi.
- BO** - Colore bordo.
- OW** - Ultimo valore assunto da X prima di un'opzione.
- ER** - Ultimo valore assunto da Y prima di un'opzione.
- E** - Eccentricità.
- P** - Colore sprite nelle opzioni "linea" e "rettangolo".
- E** - Raggio cerchio.
- AS** - Contiene le lettere della parola "disegnatore" e i valori binari letti nei Data degli sprite.
- L\$** - Stringa di caratteri grafici.
- BS** - Sprite.
- IS** - Inkey\$.

di Lino Squarza

## Uso delle label simboliche

**I**l BASIC è certamente il linguaggio più diffuso tra gli utenti di piccoli personal computer e ciò è dovuto sia alla semplicità che alla sua praticità d'uso. Il fatto di essere interpretato lo rende infatti di uso immediato anche per programmatori alle prime armi e le istruzioni che lo compongono risultano abbastanza comprensibili anche a chi non ha troppa familiarità con la lingua inglese. Quasi tutto il bene ed il male è stato scritto su questo popolare linguaggio, sulla sua lentezza di esecuzione, sulla facilità di manutenzione dei programmi ed altro, ma forse un aspetto non è stato considerato in tutta la sua importanza. Ciascuno si sarà reso conto della difficoltà di interpretare il listato di un programma composto da molti Goto o Gosub a numeri di riga astrusi ed incomprensibili senza gli adeguati commenti esplicativi. Dal momento che questi ultimi occupano spazio prezioso e tempo di battitura spesso vengono ridotti al minimo con le conseguenze ben note. Per non

parlare delle modifiche che spesso portano a imprevisti "Undef Statement Error" perché, senza saperlo, si sono cancellate o spostate istruzioni alle quali altre facevano riferimento. Nei linguaggi dotati di compilatori questo problema non sussiste perché è possibile assegnare all'indirizzo di una istruzione un nome simbolico al quale fare riferimento nelle altre parti del programma ed in modo da ricordare immediatamente la funzione svolta. Ben diverso è trovarsi di fronte ad un 1670 Gosub Calcola L'Area che ad un sibillino 1670 Gosub 3456. In effetti alcune versioni di BASIC, disponibili su macchine di classe superiore, hanno questa possibilità già implementata, a conferma della sua indiscussa utilità nella stesura e nella manutenzione dei programmi.

L'uso di label non solo numeriche, ma anche simboliche, diventa possibile anche sul C 64 semplicemente caricando la routine di gestione presentata di seguito solo con una modesta riduzione della memoria disponibile e quasi senza effetti sulla velocità di esecuzione dei programmi. Dal momento che il BASIC si trova su ROM non sembra possibile apportarvi modifiche software senza pericolosi interventi hardware sul sistema, ma ciò non è del tutto vero. Infatti una delle routine più usate durante l'esecuzione dei programmi si trova in pagina o alle locazioni RAM da \$0073 a \$008A (decimale 115-138) dove

Figura 1 - La routine Charget normale e modificata.

NORMALE	CHARGET	CHARGET MODICATA
\$0073	INC \$7A	INC \$7A
\$0075	BNE \$0079	BNE \$0079
\$0077	INC \$7B	INC \$7B
\$0079	LDA \$0201 ***	LDA \$0201
\$007C	CMP #\$3A	JMP \$9E0D ###
\$007E	BCS \$008A	NOP
\$0080	CMP #\$20	CMP #\$20
\$0082	BEQ \$0073	BEQ \$0073
\$0084	SEC	SEC
\$0085	SBC #\$30	SBC #\$30
\$0087	SEC	SEC
\$0088	SBC #\$D0	SBC #\$D0
\$008A	RTS	RTS

\*\*\* istruzione modificata continuamente  
### salto alla routine di gestione

viene messa automaticamente ad ogni accensione o Reset del sistema. Il nome è Charget e serve a prelevare dal programma o dal buffer di ingresso i caratteri che compongono le istruzioni BASIC affinché l'interprete possa eseguirle. Essa viene copiata in RAM poiché deve continuamente automodificarsi per puntare al carattere seguente cosa ovviamente impossibile in ROM. È una particolarità che si trova in tutti i dialetti BASIC di derivazione Microsoft implementati sul microprocessore 6502 e può rappresentare la via più semplice per interferire con il BASIC, al fine di aggiungervi nuovi comandi o modificare quelli esistenti, senza entrare nell'interprete vero e proprio, che continuerà a svolgere tranquillamente il suo lavoro. Se la Charget prende ogni carattere da interpretare è semplice modificarla affinché essa, prima di cederlo al BASIC, permetta ad una diversa routine di esaminarlo ed agire di conseguenza. Si può dire che potrebbero convivere due interpreti diversi, ciascuno dei quali in grado di gestire comandi diversi ed entrambi facenti capo alla stessa Charget.

Esaminando la figura 1 osserviamo che dopo la modifica ogni nuovo carattere passerà prima dalla routine che inizia all'indirizzo \$9E0D, la quale potrà poi rilasciarlo alla gestione normale tornando all'indirizzo \$0080 dopo aver eseguito le due istruzioni distrette a \$007C e \$007E; oppure sostituirsi al BASIC se di tratta di un comando diverso. Le nuove istruzioni possono essere riconosciute facendole precedere da un carattere o da una coppia di caratteri speciali di uso non frequente onde evitare ambiguità: in questo modo tutto è fatto. In effetti le cose sono però piuttosto complicate dal fatto che l'interprete fa un uso molto particolare sia dei registri di CPU, dello stack e della pagina zero, per cui è estremamente facile che si accorga della presenza di questo software estraneo reagendo molto spesso con il blocco totale del sistema o con incomprensibili malfunzionamenti. Si tratta dunque di un "trapianto" possibile, ma molto delicato e non scervo da pericoli di "rigetto". Una attenta lettura e meditazione del manuale relativo al sistema operativo del C 64 è altamente raccomandabile, nonché una mano molto abile nella programmazione Assembly.

La routine di gestione delle label simboliche innanzitutto si preoccupa di riconoscerle, grazie al simbolo che le distingue, e di rilasciarle al BASIC, fingendo che siano

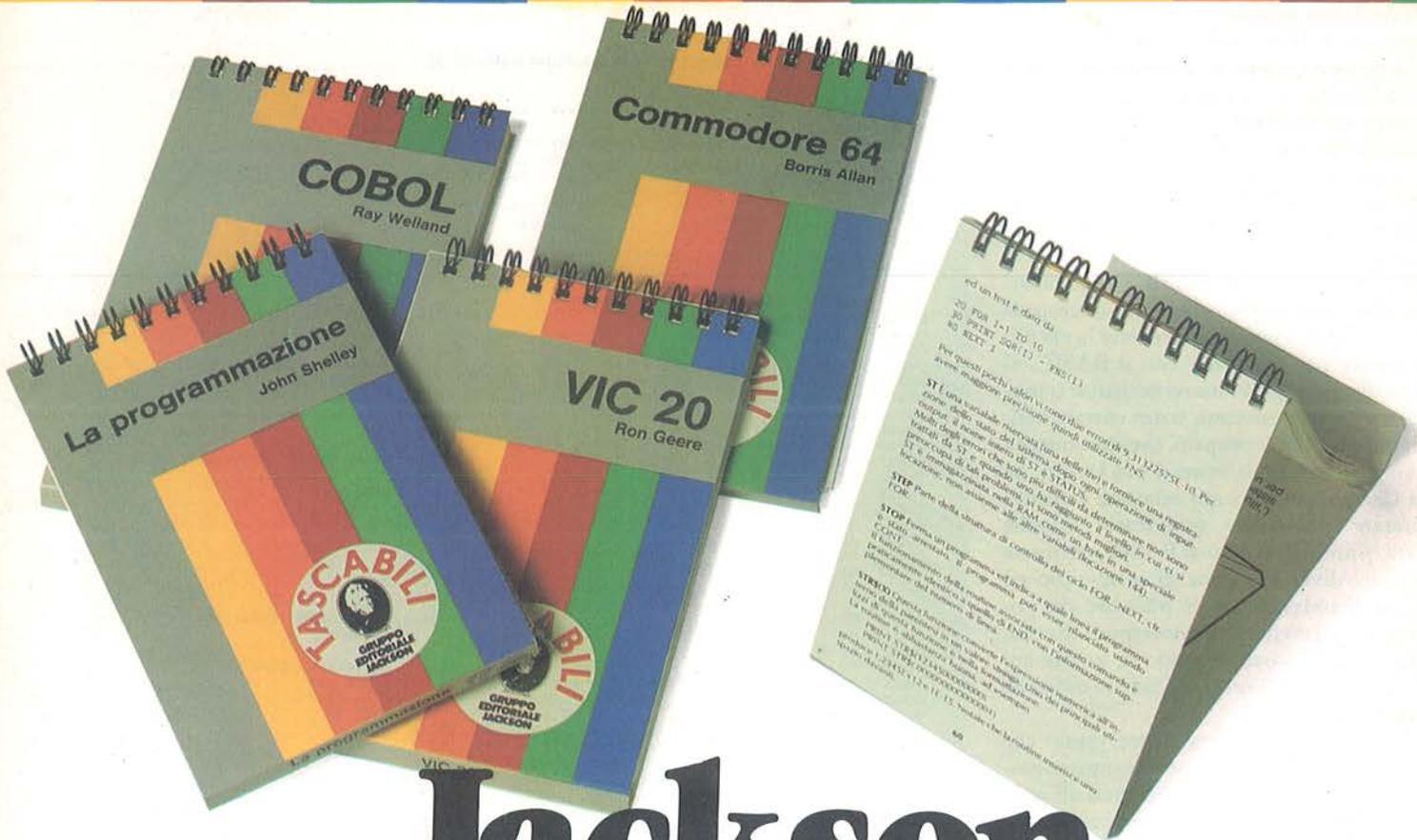
Listato 1 - Assembly della routine di gestione delle label.

```

00001 0000 ;*****
00002 0000 ;*
00003 0000 ;* PROGRAMMA PER L'USO DI *
00004 0000 ;* LABEL SIMBOLICHE IN *
00005 0000 ;* BASIC SU COMMODORE C64 *
00006 0000 ;*
00007 0000 ;* ING. L. SQUARZA -PARMA- *
00008 0000 ;* 10/84 *
00009 0000 ;*
00010 0000 ;*****
00011 0000 ;
00012 0000 ;
00013 0000 ; SIMBASIC
00014 0000 ;
00015 0000 ;
00016 0000 REM=$BF ;TOKEN PER REM
00017 0000 GOSUB=$8D ; " " GOSUB
00018 0000 GOTO=$89 ; " " GOTO
00019 0000 TOKEN=$40 ; " " LABEL
00020 0000 CHARGO=$7C ;INDIRIZZO ATTIVAZIONE ROUTINE
00021 0000 ;
00022 0000 ; *=$2B
00023 002B INPRO **+$2 ;INDIRIZZO DI INIZIO PROGRAMMA BASIC
00024 002D FIPRO **+$2 ; " " FINE "
00025 002F *=$FB ;AREA LAVORO
00026 00FB IND1 **+$2
00027 00FD CHR9 **+$1
00028 00FE COM01 **+$1
00029 00FF COM02 **+$1
00030 0100 ;
00031 0100 ; *=$14
00032 0014 ARCOM **+$2
00033 0016 *=$7A ;INDIRIZZO PROSSIMO CARATTERE
00034 007A NUCAR **+$2 ;DEL TESTO BASIC
00035 007C ;
00036 007C *=$9E00 ;INIZIO ROUTINE DI GESTIONE
00037 9E00 A9 4C START LDA #$4C ;SI ENTRA QUI PER
00038 9E02 85 7C STA CHARGO ;ATTIVARE LA GESTIONE
00039 9E04 A9 0D LDA #GESTL ;DELLE LABEL IN
00040 9E06 85 7D STA CHARGO+1 ;BASIC
00041 9E08 A9 9E LDA #GESTL ;JMP $9E0D
00042 9E0A 85 7E STA CHARGO+2
00043 9E0C 60 RTS ;FINE INIZIALIZZAZIONE
00044 9E0D ;
00045 9E0D ;
00046 9E0D 84 FE GESTL STY COM01 ;SALVA IL REG Y
00047 9E0F 48 PHA ;SALVA IL CARATTERE DA ESAMINARE
00048 9E10 A5 7B LDA NUCAR+1 ;SE E' IN RUN
00049 9E12 C9 08 CMP #08
00050 9E14 B0 0D BCS INRUN ;VA A CONTROLLARE
00051 9E16 68 PLA ;SE IN MODO DIRETTO
00052 9E17 A4 FE NOBUO LDY COM01 ;RIPRISTINA Y E ACC
00053 9E19 C9 3A CMP #3A ;SIMULA LA PARTE TOLTA
00054 9E1B B0 05 BCS EXIT ;DELLA CHARGOT ROUTINE
00055 9E1D 4C 80 00 JMP $0080 ;E RITORNA
00056 9E20 A4 FE RITOR LDY COM01
00057 9E22 60 EXIT RTS
00058 9E23 68 INRUN PLA ;RIPRISTINA L'ULTIMO CARATTERE
00059 9E24 C9 40 CMP #TOKEN ;SE DIVERSO DAL TOKEN
00060 9E26 D0 04 BNE NONAN ;POTREBBE ESSERE DA GESTIRE
00061 9E28 A9 8F LDA #REM ;ALTRIMENTI SIMULA UNA REM
00062 9E2A D0 EB BNE NOBUO ;E RITORNA AL BASIC NORMALMENTE
00063 9E2C C9 8D NONAN CMP #GOSUB ;SE E' UN GOSUB ALLORA
00064 9E2E F0 04 BEQ FORBU ;CONTINUA AD INDAGARE
00065 9E30 C9 89 CMP #GOTO ;SE NON E' NEMMENO UN GOTO
00066 9E32 D0 E3 BNE NOBUO ;NIENIE DA FARE
00067 9E34 85 FD FORBU STA CHR9 ;SALVALO
00068 9E36 A0 00 LDY #0
00069 9E38 C8 AUMY INY
00070 9E39 B1 7A LDA (NUCAR),Y ;CONTROLLA IL SUCCESSIVO
00071 9E3B F0 08 BEQ NOTRO ;SE FINE ISTRUZIONE NON CI SIAMO
00072 9E3D C9 20 CMP #' ' ;SE E' UNO SPAZIO
00073 9E3F F0 F7 BEQ AUMY ;CONTROLLA IL SUCCESSIVO
00074 9E41 C9 40 CMP #TOKEN ;SE E' PROPRIO IL TOKEN
00075 9E43 F0 04 BEQ OKVAL ;ALLORA CI SIAMO
00076 9E45 A5 FD NOTRO LDA CHR9 ;RIPRISTINA L'ULTIMO
00077 9E47 D0 CE BNE NOBUO ;E RITORNA ALLA NORMALE GESTIONE
00078 9E49 20 88 9E OKVAL JSR CERCA ;CERCA L'ISTRUZIONE CON LA LABEL
00079 9E4C B0 03 BCS SDEF ;SE C=1 ALLORA ESISTE
00080 9E4E 4C E3 AB JMP $ABE3 ;SE NO "UNDEFINED STATEMENT" E STOP
00081 9E51 68 SDEF PLA ;PREPARA UN NORMALE RITORNO
00082 9E52 68 PLA ;DALLA ROUTINE BASIC
00083 9E53 A9 A7 LDA #A7 ;DI GESTIONE DEI COMANDI
00084 9E55 48 PHA ;MODIFICANDO IL CONTENUTO
00085 9E56 A9 AD LDA #AD ;DELLO STACK
00086 9E58 48 PHA
00087 9E59 A5 FD TROVO LDA CHR9
00088 9E5B C9 8D CMP #GOSUB ;SE NON E' UN GOSUB
00089 9E5D D0 1A BNE VAIA ;VA A GESTIRE IL GOTO
00090 9E5F A9 03 LDA #3 ;CONTROLLA CHE VI SIA

```

# i Tascabili

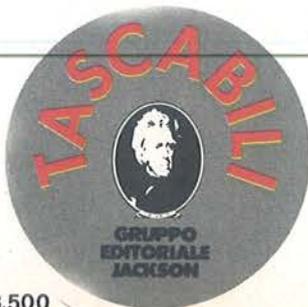


# Jackson, naturalmente.

I tascabili Jackson sono uno strumento prezioso per chi lavora con il computer.

SINCLAIR SPECTRUM cod. 017H  
VIC 20 cod. 005H  
COMMODORE 64 cod. 002H  
PC IBM cod. 018H  
APPLE IIc cod. 003H  
SHARP MZ80A cod. 014H  
LA PROGRAMMAZIONE cod. 004H  
WORD STAR cod. 008H  
UNIX cod. 009H  
LOGO cod. 020H  
MS-DOS cod. 019H  
PROGRAMMI DI STATISTICA cod. 015H  
CP/M cod. 011H  
PC-DOS cod. 012H

BASIC cod. 007H  
ASSEMBLER Z80 cod. 016H  
ASSEMBLER 6502 cod. 013H  
COBOL cod. 001H  
FORTRAN 77 cod. 010H  
PASCAL cod. 006H



OGNI TASCABILE COSTA L. 8.500

ritagliare (o fotocopiare) e spedire in busta chiusa a:  
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

### CEDOLA DI COMMISSIONE LIBRARIA

#### VOGLIATE SPEDIRMI

n° copie	codice	Titolo	Prezzo unitario	Prezzo totale
Totale				

- Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.
- Condizioni di pagamento con esecuzione del contributo spese di spedizione:
- Allego assegno della Banca  Allego fotocopia del versamento sul c/c n. 11666203 a voi intestato
- N°   Allego fotocopia di versamento su vaglia postale e voi intestato

Nome e Cognome \_\_\_\_\_

Via \_\_\_\_\_

Cap \_\_\_\_\_

Città \_\_\_\_\_

Prov. \_\_\_\_\_

Data \_\_\_\_\_

Firma \_\_\_\_\_

Spazio riservato alle Aziende Si richiede l'emissione di fattura

Partita I.V.A. \_\_\_\_\_

ORDINE  
MINIMO  
L. 50.000

## COMMODORE C 64

### Seguito programma Assembly.

```

00091 9E61 20 FB A3      JSR $A3FB      ;SPAZIO NELLO STACK
00092 9E64 A5 7B      LDA NUCAR+1   ;SALVA
00093 9E66 48         PHA           ;L'ATTUALE INDIRIZZO
00094 9E67 A5 7A      LDA NUCAR     ;NEL TESTO
00095 9E69 48         PHA           ;BASIC
00096 9E6A A5 3A      LDA $3A      ;SALVA
00097 9E6C 48         PHA           ;L'ATTUALE
00098 9E6D A5 39      LDA $39      ;NUMERO
00099 9E6F 48         PHA           ;DI RIGA
00100 9E70 A9 8D      LDA #$8D     ;METTE IL CONTRASSEGNO
00101 9E72 48         PHA           ;DI GOSUB
00102 9E73 20 79 9E   JSR VAIA     ;LO GESTISCE COME UN GOTO
00103 9E76 4C AE A7   JMP $A7AE    ;ESEGLIE LA PROSSIMA RIGA DI PROGRAMMA
00104 9E79 A0 02      LDA #2       ;GESTIONE DI UN GOTO
00105 9E7B B1 FB      LDA (IND1),Y ;NUMERO DELLA ISTRUZIONE
00106 9E7D 85 14      STA ARCOM    ;ALLA QUALE ANDARE
00107 9E7F C8         INY
00108 9E80 B1 FB      LDA (IND1),Y
00109 9E82 85 15      STA ARCOM+1
00110 9E84 20 A3 AB   JSR $ABA3    ;NORMALE GESTIONE DI UN GOTO
00111 9E87 60         RTS          ;TUTTO FATTO
00112 9E88           ;
00113 9E88           ; RICERCA DELLA RIGA BASIC
00114 9E88           ;
00115 9E88           ; CONTENENTE LA LABEL ATTUALE
00116 9E88           ; CERCA
00117 9E88 18         CLC
00118 9E89 A5 2B      LDA INPRO    ;PARTE DALL'INIZIO
00119 9E8B 85 FB      STA IND1    ;DEL TESTO BASIC
00120 9E8D 69 04      ADC #4      ;CERCA IL TOKEN ALL'INIZIO
00121 9E8F 8D E3 9E   STA GETTA+1 ;DEL TESTO
00122 9E92 A5 2C      LDA INPRO+1 ;CIOE' 4 BYTES AVANTI
00123 9E94 85 FC      STA IND1+1
00124 9E96 69 00      ADC #0
00125 9E98 8D E3 9E   STA GETTA+1
00126 9E9B 84 FF      STY COMO2  ;SALVA IL PUNTO DI INIZIO
00127 9E9D A4 FF      LDY COMO2  ;RICARICA L'INIZIO
00128 9E9F 20 E2 9E   LOPB JSR GETTA
00129 9EA2 C9 20      LOPB1 CMP #' '   ;SE NON E' SPAZIO
00130 9EA4 D0 05      BNE OKCF1  ;VA A CONTROLLARE
00131 9EA6 20 DA 9E   JSR INCR2  ;ALTRIMENTI TENTA
00132 9EA9 D0 F7      BNE LOPB1  ;COL PROSSIMO
00133 9EAB D1 7A      OKCF1 CMP (NUCAR),Y ;E' UGUALE A QUELLO CERCATO ?
00134 9EAD D0 08      BNE NURIG  ;SE NO TENTA CON LA PROSSIMA ISTRUZIONE
00135 9EAF C8         INY
00136 9EB0 20 DA 9E   JSR INCR2  ;CONTINUA COL CARATTERE SUCCESSIVO
00137 9EB3 D0 F6      BNE OKCF1  ;SE FINE ISTRUZIONE
00138 9EB5 38         OKSTR SEC    ;C=1 SIGNIFICA TROVATO !
00139 9EB6 60         RTS        ;ESCI PURE
00140 9EB7 A0 00      NURIG LDY #0    ;AVANZA DI UNA RIGA DI TESTO
00141 9EB9 B1 FB      LDA (IND1),Y ;PRENDE
00142 9EBB 48         PHA
00143 9EBD 18         CLC
00144 9EBD 69 04      ADC #4     ;L'INDIRIZZO
00145 9EBF 8D E3 9E   STA GETTA+1 ;DELLA PROSSIMA
00146 9EC2 C8         INY       ;+4 BYTE PER
00147 9EC3 B1 FB      LDA (IND1),Y ;LA RICERCA
00148 9EC5 F0 10      BEQ FINCE  ;SE ZERO
00149 9EC7 48         PHA       ;ALLORA FINE PROGRAMMA
00150 9EC8 69 00      ADC #0
00151 9ECA 8D E4 9E   STA GETTA+2
00152 9ECD 68         PLA
00153 9ECE 85 FC      STA IND1+1 ;RICORDA A QUALE
00154 9ED0 68         PLA
00155 9ED1 85 FB      STA IND1   ;RIGA E' ARRIVATO
00156 9ED3 A4 FF      LDY COMO2 ;RICERCA DALL'INIZIO DELLA LABEL
00157 9ED5 D0 C8      BNE LOPB  ;E RIPROVA
00158 9ED7 68         PLA
00159 9ED8 18         CLC
00160 9ED9 60         RTS
00161 9EDA EE E3 9E   INCR2 INC GETTA+1 ;PURTROPPO LA LABEL
00162 9EDD D0 03      BNE GETTA  ;CERCATA NON ESISTE
00163 9EDF EE E4 9E   INC GETTA+2 ;C=0 E RITORNA
00164 9EE2 AD FF FF   GETTA LDA $FFFF ;PRENDE IL CARATTERE
00165 9EE5 60         RTS        ;SUCCESSIVO DA CONTROLLARE
00166 9EE6           .END ;ISTRUZIONE MODIFICATA PER
                                ;RICERCA VELOCE
    
```

```

SYMBOL TABLE
SYMBOL VALUE
ARCOM 0014  AUMY  9E38  CERCA 9E88  CHARGO 007C
CHR  00FD  COMO1 00FE  COMO2 00FF  EXIT  9E22
FINCE 9ED7  FIPRO 002D  FORBU 9E34  GESTL 9E0D
GETTA 9EE2  GOSUB 008D  GOTO  0089  INCR2 9EDA
IND1  00FB  INPRO 002B  INRUN 9E23  LOPB 9E9F
LOPB1 9EA2  LOPY  9E9D  NOBUO 9E17  NONAN 9E2C
NOTRO 9E45  NUCAR 007A  NURIG 9EB7  OKCF1 9EAB
OKSTR 9EB5  OKVAL 9E49  REM  00BF  RITOR 9E20
SIDEF 9E51  START 9E00  TOKEN 0040  TROVO 9E59
VAIA  9E79
END OF ASSEMBLY
    
```

commenti onde evitare fatali "Syntax Error". Quando invece si trova di fronte a Goto @ o Gosub @ va alla ricerca della label indicata e punta alla normale routine dell'interprete dopo aver sostituito al nome il numero di riga trovato. Naturalmente al di fuori di questi casi ritorna alla Charged come se nulla fosse successo. L'esecuzione delle altre istruzioni non viene minimamente alterata e rimangono quindi ancora possibili i Gosub e Goto numerici. Per chi volesse saperne di più, o apportare qualche modifica, è disponibile il listato 1 commentato della routine di gestione in Assembly. L'occupazione effettiva di memoria va da \$9E00 a \$9EE6 per cui rimane spazio sino a \$9FFF per ulteriori routine di utilità. Volendo si potrebbe rilocare tutto a \$9F00 perdendo così solo 256 byte di RAM. Chi volesse usare un diverso carattere per il riconoscimento delle label potrà sostituire solo la direttiva Token=\$40 con il valore ASCII del simbolo scelto e riassemble tutto. Volendo modificare direttamente in RAM basterà porre il nuovo valore agli indirizzi \$9E25 e \$9E42.

Il programma BASIC di caricamento (listato 2) dovrà essere caricato come primo ed eseguito immediatamente. Da quel momento si potranno utilizzare programmi con label simboliche. Attenzione ai valori Data perché sono tanti, ed è molto facile commettere errori di battitura con le conseguenze purtroppo ormai note.

Per chiarezza viene fornito anche un semplice programma (listato 3) di calcolo dell'area del triangolo che fa uso di tutte le nuove possibilità offerte dall'uso di label esplicative. Esse possono avere qualunque lunghezza inferiore a 255 caratteri sempre però precedute dall'indicatore @ e comprendere spazi. Si è rivelato molto utile spostarle verso destra di alcuni spazi affinché siano facilmente visibili durante la lettura del programma, ma attenzione a non aggiungerne all'interno delle label stesse, altrimenti diventerebbero irriconoscibili alla routine di gestione. Una piccola limitazione per finire. Una label non può essere la prima istruzione di un programma, d'altro canto è sempre possibile farla precedere da una Rem quindi la cosa non è poi così drammatica.

### Listato 2 - Il programma caricatore BASIC.

```

10 GOSUB1000
11 REM
20 PRINT"{CLR}":PRINT"{ 11 GIU' }
  { 11 DES}CARICATO !!"
28 REM
29 REM   ATTIVAZIONE
30 SYS ZZ:CLR:NEW
35 REM   PROCEDURA ATTIVATA
40 END
1000 REM   PROGRAMMA IN LINGUAGGIO
1001 REM   MACCHINA DA CARICARE DA
1002 REM       $9E00
1003 REM
1005 DATA169,76,133,124,169,13,133,125,16
  9,158
1010 DATA133,126,96,132,254,72,165,123,20
  1,8
1020 DATA176,13,104,164,254,201,58,176,5,
  76
1030 DATA128,0,164,254,96,104,201,64,208,
  4
1040 DATA169,143,208,235,201,141,240,4,20
  1,137
1050 DATA208,227,133,253,160,0,200,177,12
  2,240
1060 DATA8,201,32,240,247,201,64,240,4,16
  5
1070 DATA253,208,206,32,136,158,176,3,76,
  227
1080 DATA168,104,104,169,167,72,169,173,7
  2,165
1090 DATA253,201,141,208,26,169,3,32,251,
  163
1100 DATA165,123,72,165,122,72,165,58,72,
  165
1110 DATA57,72,169,141,72,32,121,158,76,1
  74
1120 DATA167,160,2,177,251,133,20,200,177
  ,251
1130 DATA133,21,32,163,168,96,24,165,43,1
  33
1140 DATA251,105,4,141,227,158,165,44,133
  ,252
1150 DATA105,0,141,227,158,132,255,164,25
  5,32
1160 DATA226,158,201,32,208,5,32,218,158,
  208
1170 DATA247,209,122,208,8,200,32,218,158
  ,208
1180 DATA246,56,96,160,0,177,251,72,24,10
  5
1190 DATA4,141,227,158,200,177,251,240,16
  ,72
1200 DATA105,0,141,228,158,104,133,252,10
  4,133
1210 DATA251,164,255,208,200,104,24,96,23
  8,227
1220 DATA158,208,3,238,228,158,173,255,25
    
```

### Seguito listato caricatore BASIC.

```

  5,96
1245 REM   CALCOLA $9E00 IN DECIMALE
1250 ZZ=9*16↑3+14*16↑2
1251 REM
1252 REM   DIVIDE IN PARTE ALTA E BASSA
1260 ZH=INT(ZZ/256):Z1=ZZ-ZH*256
1265 REM
1267 REM   SPOSTA IN BASSO IL TOP
1268 REM   DELLA MEMORIA UTILE AL BASIC
1270 POKE51,Z1:POKE52,ZH:POKE55,Z1:POKE56
  ,ZH
1305 REM
1306 REM   CARICA LA ROUTINE
1310 FORK=ZZ TO ZZ+229
1320 READ X
1330 POKE K,X
1340 NEXT K
1345 REM   OK TUTTO PRONTO
1350 RETURN
    
```

### Listato 3 - Un programma d'esempio sull'uso della label.

```

1 REM   UTILIZZO LABEL
2 REM   NEI PROGRAMMI BASIC
3 REM
4 REM   CALCOLO AREA TRIANGOLO
5 REM
6 REM   INIZIO PROGRAMMA
100 @DA CAPO
110 PRINT "{CLR}"
120 GOSUB   @INPUT BASE
130 GOSUB   @INPUT ALTEZZA
140 GOSUB   @CALCOLA L'AREA
150 PRINT "AREA = ";AREA
160 INPUT "CONTINUO (S/N) ";SI$
170 IF SI$="N" THEN 190
180 GOTO   @DA CAPO
190 END
200 @INPUT BASE
210 INPUT "BASE = ";BASE
220 RETURN
300 @INPUT ALTEZZA
310 INPUT "ALTEZZA = ";ALTEZZA
320 RETURN
400 @CALCOLA L'AREA
410 AREA=(BASE*ALTEZZA)/2
420 RETURN
    
```

# SERVIZIO SOFTWARE

# Bit

Bit e Personal Software propongono ai propri lettori i dischi o le cassette dei programmi pubblicati. Uso e descrizione dei programmi si trovano sui rispettivi numeri delle riviste.



# PERSONAL SOFTWARE

Bit n°	Programma	Sistema	Codice	Supporto	Bit n°	Programma	Sistema	Codice	Supporto
38	Gioco della scimmia Spaccamattoni	VIC 20	VI381A	Cassetta	63	Operazioni in pagina grafica	Apple II	AP632A	Disco
38	Planet	Apple II	AP382B	Disco	63	Grafici a barre verticali affiancate	C 16	C1631B	Cassetta
42	Apple-Chef	Apple II	AP422A	Disco	63	Contabilità casalinga	C 64	C6632C	Disco
42	Provariflessi	VIC 20	VI421B	Cassetta	63	Disegnare sullo schermo con un MSX	MSX	MS631D	Cassetta
45	Tiny FORTH	Apple II	AP452A	Disco	63	Over BASIC	ZX Spectrum	SP631E	Cassetta
45	Ali Baba	ZX Spectrum	SP451B	Cassetta	63	Le otto Regine	TI99/4A	TI631F	Cassetta
46	Forzaquattro	Apple II	AP462A	Disco	63	Egitto 2000	VIC 20	VI631G	Cassetta
48	Simulavolo	ZX Spectrum	SP481A	Cassetta					
48	Memory Alfa IV	C 64	C6481B	Cassetta					
49	Scorpion	Apple II	AP492A	Disco					
50	Fp-Plot	Apple II	AP502A	Disco					
50	Prima e Terza	ZX Spectrum	SP501B	Cassetta					
51	Magicalcatalog	Apple II	AP512A	Disco					
53	Partita a golf	VIC 20	VI531A	Cassetta					
53	Analisi numerica	C 64	C6531B	Cassetta					
53	PL/Bit: il compilatore	Apple II	AP532C	Disco					
54	Constellations	Apple II	AP542A	Disco					
54	Come polarizzare i transistor col C 64	C 64	C6541B	Cassetta					
58	Memory Omega I	C 64	C6582A	Disco					
58	Copy disk per C 64	C 64	C6582B	Disco					
59	Checksum 64	C 64	C6592A	Disco					
59	Checksum 64	C 64	C6591B	Cassetta					
59	Data-Bank	ZX Spectrum	SP591C	Cassetta					
60	Life HGR	Apple II	AP602A	Disco					
60	Tutti pittori	C 64	C6601B	Disco					
60	Difesa della Terra	C 16	C1601C	Cassetta					
60	Lost on the pack	Sega	SE601D	Cassetta					
60	Setdata e Wordproc	ZX Spectrum	SP601E	Cassetta					
60	Il Barone Rosso	TI99/4A	TI601F	Cassetta					
60	Word processor	C 64	C6602G	Disco					
60	Othello	VIC 20	VI601H	Cassetta					
61	Disegno di mobili componibili	C 16	C1611A	Cassetta					
61	Esperimento di Millikan	C 64	C6611B	Cassetta					
61	Esperimento di Millikan	C 64	C6612C	Disco					
61	Disegnare in alta risoluzione	ZX Spectrum	SP611D	Cassetta					
61	Printing music	TI99/4A	TI611E	Cassetta					
61	Musica facile	Sega	SE611F	Cassetta					
61	VIC-Calcul	VIC 20	VI611G	Cassetta					
62	Gestione conto corrente	C 64	C6622A	Disco					
62	Gioco della tombola	TI99/4A	TI621B	Cassetta					
62	Aspc: lo Spectrum contro la carie	ZX Spectrum	SP621C	Cassetta					

P.S. n°	Programma	Sistema	Codice	Supporto
3	La carta del cielo	Apple II	AP032A	Disco
4	Collisione	Apple II	AP042A	Disco
5	Interi in precisione multipla	Apple II	AP052A	Disco
7	Pretty printer	Apple II	AP072A	Disco
14	Shape table	Apple II	AP072A	Disco
19	Data base modulare	Apple II	AP072A	Disco
20	Tool-Kit	C 64	C6141A	Cassetta
30	Type-Writer	VIC 20	VI192A	Disco
	Scopa	C 64	C6201A	Cassetta
	Geo-Race	ZX Spectrum	SP301A	Cassetta

Tutti i dischi e le cassette dei programmi sono in vendita a L. 15.000 ciascuno.

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando  
**Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano**

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su Bit - Personal Software

Cod. <input type="text"/>	a L. <input type="text" value="15000"/>
Cod. <input type="text"/>	a L. <input type="text"/>
Cod. <input type="text"/>	a L. <input type="text"/>
Cod. <input type="text"/>	a L. <input type="text"/>
<b>+ SPESE POSTALI (contributo fisso)</b>	<input type="text" value="3000"/>

**TOTALE L.**

che pagherò al postino alla consegna del pacco



**GRUPPO EDITORIALE JACKSON**

Cognome.....

Nome.....

Indirizzo.....

CAP.....

Città.....

Firma.....

di Martino Sangiorgio

## Caratteri & colori

**M**olti possessori di MZ-700 si saranno chiesti come mai, avendo tale computer la possibilità di visualizzare solo 8 colori, nel primo dei programmi dimostrativi in dotazione si possono vedere tutta una serie di colori non dichiarata nel manuale. In più, alcuni avranno anche avuto delle difficoltà nel comprendere l'utilizzo del secondo set di caratteri grafici. Ma andiamo in ordine.

Lo Sharp MZ-700 dispone di una RAM video di 4 Kbyte. I primi 2 Kbyte (dall'indirizzo \$D000 all'indirizzo \$D7FF) vengono utilizzati per memorizzare i caratteri da visualizzare, mentre i successivi 2 Kbyte (dall'indirizzo \$D800 all'indirizzo \$DFFF) vengono utilizzati per memorizzare, per ognuno dei caratteri presenti nella prima zona della RAM video, le corrispondenti informazioni sui colori di sfondo e di primo piano.

I caratteri vengono memorizzati nella prima zona della RAM video non con la loro codifica ASCII, ma con una codifica particolare. Tale codifica, detta "di visualizzazione" o "di schermo", è formata da due gruppi di 256 caratteri, visibili alle pagine 157 e 159 del manuale. La scelta del primo o del secondo set di caratteri dipende dalle informazioni relative al colore, come vedremo più avanti.

Di norma, a fronte, ad esempio, di una istruzione Print, vengono innescate delle subroutine del programma Monitor che convertono i codici ASCII in codici di schermo. Comunque, essendo anch'essa un'area RAM, è possibile modificare la RAM video, per esempio con delle Poke, in modo da visualizzare qualsiasi carattere voluto.

E veniamo ora al colore. A pagina 133 del manuale è riportata la struttura del byte che contiene le informazioni sul colore. Da tale struttura si nota che i primi tre bit (da D0 a D2) sono utilizzati per il colore di sfondo del carattere, che il quarto bit (D3) non è usato, che i successivi tre bit (da D4 a D6) sono utilizzati per il colore di primo piano del carattere, e che il bit più significativo (D7) non è utilizzato.

Semberebbe quindi di avere a disposizione solo 64 combinazioni di colore (8 di sfondo per 8 di primo piano), e che i due bit non utilizzati (il quarto e l'ottavo) non abbiano altro effetto che ripetere combinazioni di colore già verificate.

Per controllare ciò utilizziamo il programma 1. Esso riempie 11 righe del video (dall'indirizzo \$D028 all'indirizzo \$D1DF della RAM video) con il carattere pieno (codice di schermo \$43, vedere a pagina 157 del manuale), lasciando vuota la zona sottostante. Imposta poi il colore sia delle 11 righe precedenti, sia delle 11 righe sottostanti (cioè dall'indirizzo \$D828 all'indirizzo \$DB97), variando il colore impostato da 0 a 255. In tal modo il video viene suddiviso in due finestre, ognuna di 11 righe: nella finestra superiore verrà visualizzato il colore di primo piano, mentre in quella inferiore il colore di sfondo. Alla fine di ognuna delle 256 combinazioni di colore verrà chiesto se si vuole proseguire, in modo da dare tempo, a chi lo volesse, di prendere appunti.

Eseguiamo ora il programma. Col colore 0 si avranno 22 righe nere (carattere nero su sfondo nero), col colore 1 si avranno le prime 11 righe nere e le successive blu (carattere nero su sfondo blu) e così via. Per i primi 8 valori del colore (da 0 a 7) si avrà il nero per il carattere e tutti gli 8 colori per lo sfondo.

Dal valore 8 al 15 verranno ripetute le combinazioni di colore appena viste (carattere nero sugli 8 colori di sfondo), e ciò dimostra che il quarto bit (D3) del byte del colore è effettivamente inutilizzato. Dal valore 16 al 23 si avrà il blu come carattere e gli otto colori come sfondo, dal 24 al 31 la ripetizione del blu come carattere, e così via, fino al valore 127 del colore, con l'ordinamento dei colori visibile a pagina 79 del manuale (Nero, Blu, Rosso, Porpora, Verde, Azzurro, Giallo, Bianco).

Al valore 128 di colore si vede ancora il nero del carattere e il nero di sfondo, e sembrerebbe quindi iniziata la ripetizione delle prime 128 combinazioni di colore. Ma non è proprio così. Infatti, al valore 129, corrisponde ancora, come al valore 1, il nero di primo piano e il blu di sfondo, ma è cambiato il carattere visualizzato nelle prime 11 righe. Esso, infatti, non è più il carattere pieno, ma è un altro carattere grafico particolare. Lo andiamo a cercare sul manuale e lo troviamo, con la codifica \$43, nel secondo set di caratteri grafici, a pagina 159. Questo dimostra che il bit più

significativo del byte del colore è effettivamente inutilizzato per il colore (infatti, da 128 a 255 si avranno le stesse combinazioni di colore viste per i valori da 0 a 127), ma permette di evidenziare i caratteri presenti nel secondo set grafico.

In sostanza, se stiamo evidenziando un carattere del primo set grafico, con un determinato colore, basta aggiungere 128 al byte del colore per evidenziare il carattere, con uguale codifica, del secondo set grafico, tenendo invariati i colori di sfondo e di primo piano.

Ma non è ancora finita.

Abbiamo visto che i colori utilizzabili sono effettivamente solo 8, il che non spiega ancora tutti i colori visibili nel programma dimostrativo. Digitiamo allora il programma 2. Esso è molto simile al precedente. Riempie 22 righe del video con il carattere \$EF (vedere a pagina 157 del manuale), e imposta quindi un colore da 0 a 255 nella stessa zona del video.

Eseguiamo il programma. Al valore 00 avremo ancora il video nero, ma già al valore 1 vediamo il video colorarsi di un blu diverso dal solito, più scuro. Anche il rosso (valore 2 di colore) è più scuro del solito.

Come si spiega ciò? La chiave di tutto è la particolare conformazione del carattere che abbiamo utilizzato, fatta di 8 punti di sfondo di uguale dimensione. A causa della particolarità del televisore (il monitor, più preciso, non permette completamente ciò), i colori del primo piano e dello sfondo sembrano fondersi, originando un colore diverso da entrambi.

Proseguite nell'elaborazione del programma 2 e vedrete quante altre combinazioni otterrete, alcune molto piacevoli, altre un po' meno. Annotatevi le migliori (compreso il valore del colore che le genera), in modo da poterle usare in futuro, ad esempio per comporre titoli colorati per i vostri programmi. Possiamo ora scrivere ed eseguire il programma 3. Esso permette di riempire 22 righe del video col carattere voluto e di assegnargli il colore voluto. Potremo quindi verificare tutto quanto visto finora. ■

## SHARP MZ-700

Listato 1 - Visualizzazione delle combinazioni dei colori di primo piano e di sfondo dei caratteri.

```

10 PRINT "0"
20 FOR J=$D028T0$D397:POKEJ,$EF:NEXTJ
30 FOR I=0T0255
40 CURSOR 0,0:PRINT"COLORE ";I
50 FOR J=$D828T0$DB97:POKEJ,I:NEXTJ
60 CURSOR 0,24:PRINT"VUOI CONTINUARE?";
70 GET C$:IF C$="" GOTO 70
80 IF C$="N" GOTO110
90 IF C$<>"S"GOTO70
100 NEXT I
110 END
    
```

Listato 2 - Visualizzazione di tutta la gamma di colori ottenibili con l'utilizzo del carattere \$EF.

```

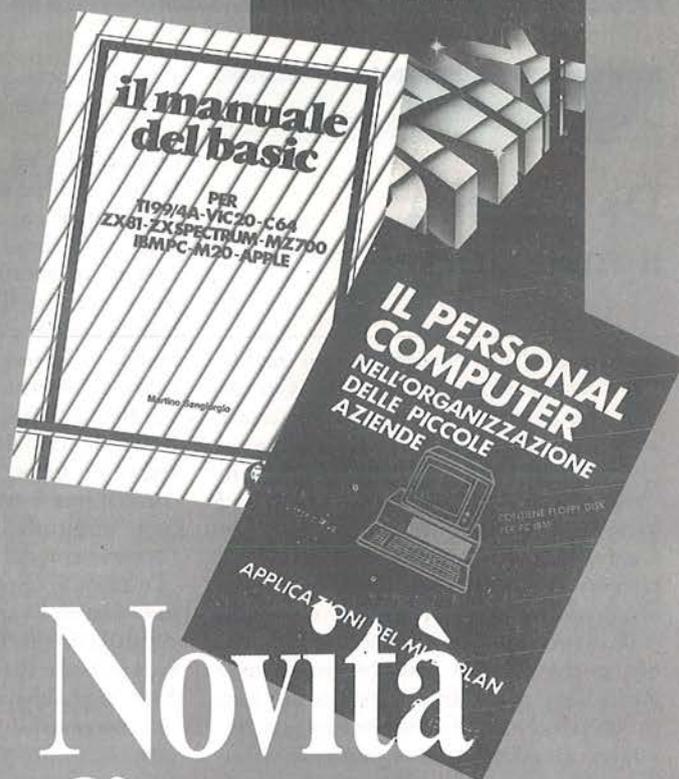
10 PRINT "0"
20 FOR J=$D028T0$D1DF:POKEJ,$43:NEXTJ
30 FOR I=0T0255
40 CURSOR 0,0:PRINT"COLORE ";I
50 FOR J=$D828T0$DB97:POKEJ,I:NEXTJ
60 CURSOR 0,24:PRINT"VUOI CONTINUARE?";
70 GET C$:IF C$="" GOTO 70
80 IF C$="N" GOTO110
90 IF C$<>"S"GOTO70
100 NEXT I
110 END
    
```

Listato 3 - Visualizzazione del carattere voluto con la combinazione di colori scelta.

```

10 PRINT "0"
20 CURSOR 0,0:PRINT"CARATTERE":CURSOR20,0:PRINT"COLORE"
30 CURSOR 12,0:INPUTCA$:IF(VAL(CA$)<0)+(VAL(CA$)>255)GOTO30
40 CURSOR 28,0:INPUTCL$:IF(VAL(CL$)<0)+(VAL(CL$)>255)GOTO40
50 FOR J=$D028T0$D397:POKEJ,VAL(CA$):NEXT J
60 FOR J=$D828T0$DB97:POKEJ,VAL(CL$):NEXT J
70 CURSOR 0,24:PRINT"VUOI CONTINUARE?";
80 GET C$:IF C$="" GOTO 80
90 IF C$="N" GOTO120
100 IF C$<>"S"GOTO80
110 GOTO30
120 END
    
```

Ambiente  
Unix



# Novità firmate Jackson.

IL PERSONAL COMPUTER  
NELL'ORGANIZZAZIONE DELLE PICCOLE  
AZIENDE: APPLICAZIONI DEL MULTIPLAN  
COD. 578P L. 29.000

IL MANUALE DEL BASIC  
COD. 534A L. 45.000

AMBIENTE UNIX  
COD. 543P L. 19.000



GRUPPO  
EDITORIALE  
JACKSON

La biblioteca che fa testo.

# I SEGRETI DEI PERSONAL

di Marcello Spero

## Da cassetta a microdrive

I microdrive consentono un rapido accesso a programmi e dati. Ciò che prima richiedeva minuti per essere caricato, si rende ora disponibile nel giro di qualche secondo. Chi abbia provato i microdrive difficilmente si rassegna ad usare le cassette.

Il software commerciale disponibile direttamente su microdrive, d'altra parte, è praticamente inesistente, per il costo e la difficoltà di produzione. Come se non bastasse, ben poche sono le case disposte a dare indicazioni per il trasferimento dei loro prodotti, nel timore di incrementare la "pirateria" (argomentazione piuttosto debole, visto che sono sufficienti due registratori per ottenere tutte le copie che vogliamo). Fanno eccezione le case produttrici di software "serio" (Tasman, Campbell, Hisoft), che forniscono tutte le informazioni necessarie.

Perché sono necessarie informazioni da parte del produttore, per il trasferimento di un programma da cassetta a microdrive? Perché la gran parte del software, di fatto dà, durante questa operazione, parecchi problemi, per comprendere i quali è neces-

sario prendere brevemente in considerazione la meccanica di funzionamento dei microdrive.

Osservando, sul manuale dello Spectrum, la mappa di memoria, notiamo due aree, denominate rispettivamente "Microdrive map" e "Channel information". Con l'inserimento di Interfaccia 1 e microdrive, queste aree si "gonfiano" di 32 e 595 byte. La mappa contiene informazioni sulla distribuzione dei vari blocchi (di 12 byte ciascuno) all'interno della cartuccia, mentre i 595 byte di informazioni comprendono un buffer di 512 byte (in grado quindi di contenere un blocco) ed alcuni altri dettagli, come il nome ed il tipo del file e così via (per maggiori informazioni consultate l'appendice 3 del manuale dell'interfaccia).

Durante le operazioni che coinvolgono i microdrive, i vari blocchi vengono trasferiti dalla cartuccia alla memoria o viceversa, a seconda che sia in corso un Load o un Save. Il problema è che, per creare lo spazio necessario, programma e variabili eventualmente presenti in memoria devono essere spostate in avanti di  $32 + 595 = 627$  byte. Inoltre, tanto per complicare le cose, l'Interfaccia 1 aggiunge 58 nuove variabili all'area delle variabili di sistema. In conclusione, il programma BASIC viene a trovarsi ad un indirizzo di memoria maggiore di circa 685 byte rispetto al normale. Ma perché questo costituisce un problema?

I blocchi di linguaggio macchina che normalmente andrebbero a finire subito al di sopra del BASIC, rischiano in questa situazione di sovrapporsi al programma o,

peggio ancora, di "sporcare" parte della mappa del microdrive o dell'area dei canali. In ambedue i casi il sistema andrà in "crash" mentre il microdrive sta girando, costringendoci a togliere l'alimentazione con il rischio di danneggiare le informazioni contenute sulla cartuccia.

I programmi possono essere suddivisi in tre categorie. Quelli facili da trasferire su microdrive, poiché non occupano col linguaggio macchina indirizzi di memoria troppo bassi. Ci sono poi quei programmi che richiedono di essere caricati ad un indirizzo più elevato di quello a cui è previsto che funzionino, e in un secondo tempo spostati al loro corretto indirizzo. L'ultima categoria, infine, comprende i programmi che, pur richiedendo il medesimo procedimento di caricamento e trasferimento, sono talmente lunghi da non poter essere caricati interamente; questo perché hanno bisogno anche di quei 700 byte utilizzati per il funzionamento del microdrive, nonché dello spazio occupato dalla routine di trasferimento. In questo caso si dovrà ricorrere ad un caricamento a blocchi, normalmente due, che verranno trasferiti uno alla volta. La prima categoria comprende i programmi scritti interamente in BASIC (fermateli con Break e quindi salvateli in modo Auto-Start sulla cartuccia) e quelli che, pur contenendo blocchi di linguaggio macchina, utilizzano indirizzi elevati per essere conservati. Si tratta in pratica di tutti i blocchi che hanno come indirizzo di inizio valori superiori a 25.000, con eventuali differenze a seconda delle dimensioni del-

Listato 1 - Lettore di header.

```
1 REM *****
2 REM *
3 REM *   PROGRAMMA 1   *
4 REM *
5 REM *   lettura header *
6 REM *
7 REM *****
8 REM
10 BORDER 0: PAPER 0: INK 7
20 RESTORE
30 FOR i=USR "o" TO USR "o"+18
40 READ a: POKE i,a: NEXT i
50 POKE USR "o"+5,PEEK 23675
60 POKE USR "o"+6,PEEK 23676
70 DIM t$(4,5)
80 FOR i=1 TO 4
90 READ t$(i): NEXT i
100 LET o=USR USR "o"
110 IF PEEK USR "b"=0 THEN GO TO
0 100
```

```
120 IF o=0 THEN PRINT "ERRORE DI CARICAMENTO": GO TO 21
130 PRINT "Nome : ";
140 FOR i=USR "a"+1 TO USR "b"+
2
150 PRINT CHR$( PEEK i);
160 NEXT i
170 PRINT "Tipo : ";t$(PE
EK USR "a"+1)
180 LET b=USR "b"
190 PRINT "Inizio : ";PEEK (b
+5)+256*PEEK (b+6)
200 PRINT "Lunghezza: ";PEEK (b
+3)+256*PEEK (b+4);" byte"
210 POKE 23692,255
220 POKE USR "b",0
230 GO TO 100
1000 DATA 55,62,0,221,33,0,0,17,
17,0,205,86,5,1,1,0,216,11,201
1010 DATA "Basic","Data","Data",
"Bytes"
```

## ZX SPECTRUM

l'eventuale programma BASIC di caricamento. Il programma 1 dà parecchie informazioni utili su ciò che è registrato su cassetta: il numero di linea dell'Auto-Start, per i programmi BASIC, l'indirizzo di inizio e la lunghezza per i blocchi di linguaggio macchina.

Per utilizzarlo, date Run e quindi fate partire il nastro che vi interessa, al normale volume di caricamento. Per ciascun blocco annotatevi, o fate stampare, tutti i dettagli. Un output tipico è:

**Nome:** Invaders

**Tipo:** byte

**Start:** 16.384

**Lunghezza:** 6.912 byte

**Nome:** Invaders

**Tipo:** byte

**Start:** 32.768

**Lunghezza:** 32.768 byte

Questo può essere considerato il formato normale: programma BASIC di caricamento, schermata introduttiva e blocco di linguaggio macchina.

Dopo aver dato New caricate con Merge il BASIC. Il suo contenuto sarà qualcosa del tipo:

```
10 CLEAR 32767
20 LOAD ""CODE:LOAD"" CODE
30 PRINT USR 35000
```

Per quanto riguarda la schermata introduttiva, sarà meglio lasciarla perdere; questo tipo di introduzione ha infatti lo scopo di "intrattenere" l'utilizzatore per il tempo

(molto) necessario al caricamento del blocco in linguaggio macchina che rappresenta il programma vero e proprio. Il problema dell'attesa non esiste con il microdrive, e d'altra parte lo spazio sulla cartuccia è prezioso (o sarebbe meglio dire costoso?). Correggete perciò il BASIC in questo modo:

```
10 CLEAR 32767
20 LOAD ★"M";1;"Invaders.c" CODE
30 PRINT USR 35000
```

Notate come sia stato necessario cambiare il nome del blocco in linguaggio macchina, poiché non è possibile avere su di una stessa cartuccia due file con lo stesso nome. Salvate il BASIC così modificato in modo che vada in Auto-Run dalla linea 10. Quindi date:

```
CLEAR 32767:LOAD "" CODE
```

per caricare il linguaggio macchina dalla cassetta. Ora salvatelo sulla cartuccia dando, ad esempio:

```
SAVE ★"M";1;"Invaders.c" CODE
32768,32768
```

A questo punto reinizializzate il sistema con:

```
RANDOMIZE USR 0
```

e provate a caricare tutto il programma dal microdrive. In caso di problemi, ricontrollate gli indirizzi e la lunghezza.

Una strategia diversa è necessaria per i programmi con blocchi di linguaggio mac-

china che iniziano al di sotto dell'indirizzo 25000 (pressappoco). Immaginate ad esempio una situazione in cui il blocco in linguaggio macchina sia composto da 900 byte, ed inizi all'indirizzo 24000. Caricarlo così com'è da microdrive significherebbe finire sopra l'area dei canali ed il BASIC.

La soluzione consiste nel caricarlo ad un indirizzo più elevato, e quindi spostarlo in basso al suo indirizzo corretto. Il programma 2 fa proprio questo, servendosi della potente e velocissima istruzione in linguaggio macchina Ldir.

Per proseguire con lo stesso esempio, la prima cosa da fare sarà caricare il blocco dalla cassetta all'indirizzo, ad esempio, 30000. Fate girare il programma 2, e dategli i seguenti valori:

- inizio blocco: 30.000;
- lunghezza blocco: 9.000;
- nuovo indirizzo: 24.000;
- indirizzo di esecuzione: 24.000.

L'indirizzo di esecuzione può essere trovato esaminando il blocco BASIC del programma stesso; sarà l'indirizzo cui si riferisce l'istruzione ...Usr...

Il programma 2 pone all'indirizzo "inizio + lunghezza", nel nostro caso 39000 (3000+9000), una routine di copia lunga 14 byte. Salvate tutto quanto (linguaggio macchina e routine) su cartuccia con un comando tipo:

```
SAVE ★"M";1;"Nome file" CODE
30000,9014
```

Notate come la lunghezza sia stata aumen-

Listato 2 - Programma per il trasferimento di blocchi di linguaggio macchina.

```
10 REM *****
20 REM *          PROGRAMMA 2          *
30 REM *          copia di blocco      *
40 REM *          *****             *
50 REM *****
60 DEF FN l(x)=x-256*INT (x/256)
70 DEF FN h(x)=INT (x/256)
80 INPUT "Inizio blocco : "
90 ;inizio
100 INPUT "Lunghezza blocco : "
110 ;lungh
120 INPUT "Nuovo indirizzo : "
130 ;dest
140 INPUT "Indirizzo di esec. : "
150 ;esec
```

```
100 LET startrt=inizio+lungh
110 LET x=0
120 READ v
130 IF v=999 THEN PRINT ""Indirizzo inizio routine: ";startrt
140 STOP
150 POKE startrt+x,v
160 LET x=x+1
170 GO TO 120
1000 DATA 33, FN l(inizio), FN h(inizio), 1, FN l(lungh), FN h(lungh), 17, FN l(dest), FN h(dest), 237, 175, 195, FN l(esec), FN h(esec)
1010 DATA 999
```

tata di 14 byte per salvare blocco e routine.

Modificate quindi il BASIC per poter caricare da microdrive il linguaggio macchina, e l'indirizzo di esecuzione (...Usr nnnnn) affinché sia quello della routine di copia. Salvatelo sulla cartuccia in modo Auto-Run e quindi reinizializzate il sistema e provate a caricare.

E per i blocchi troppo grossi che non possono essere spostati con la tecnica appena vista? Supponiamo, ad esempio, di aver un blocco di linguaggio macchina lungo 41.535 byte con inizio dall'indirizzo 24000, e che quindi non lascia lo spazio necessario ai microdrive.

Ciò che dobbiamo fare in questo caso è spezzare il blocco in due, caricare la parte superiore al suo corretto indirizzo, la parte bassa del Display File ed utilizzare la routine di copia del programma 2 per trasferire la parte inferiore al suo corretto indirizzo, dopo che tutte le operazioni con i microdrive siano terminate. Per proseguire con l'esempio, date prima di tutto:

```
CLEAR 23999: LOAD "" CODE
```

Copiate quindi i primi 1.000 byte nel Display File, usando un breve programma tipo:

```
10 FOR x=0 TO 999
20 POKE 16386+x,PEEK(24000+X)
30 NEXT x
```

dopodiché salvateli in un file temporaneo su nastro con:

```
SAVE "temp" CODE 16386,1000
```

Date ora un CLEAR 24999 per creare lo spazio per il funzionamento dei microdrive, e salvate la parte superiore del blocco su cartuccia con:

```
SAVE *M;1;"Parte alta" CODE
25000,40535
```

Reinizializzate il sistema, caricate il programma 2 e fatelo girare con Goto 10 e non con Run, che cancellerebbe il Display File, dandogli questi dati:

- inizio blocco: 16.386;
- lunghezza blocco: 1.000;
- nuovo indirizzo: 24.000;
- indirizzo di esecuzione: 24.000.

Salvate il tutto su microdrive con:

```
SAVE * "M";1;"Parte bassa" CODE
16386,1014
```

Notate ancora una volta come la lunghezza sia stata aumentata di 14 per includere la routine di copia. Infine, modificate il BASIC affinché carichi "Parte alta" e "Parte bassa" da microdrive prima di eseguire:

```
PRINT USR 17386
```

Salvatelo sulla medesima cartuccia, reinizializzate il sistema e, dopo aver fatto i debiti scongiuri, provate a ricaricare. Un avvertimento riguarda lo stack. Controllate sempre la sua posizione, poiché se venisse sovrascritto da qualcosa il "crash" del sistema sarebbe certo. La posizione dello stack è comunque controllata dall'istruzione Clear, e questo può evitare molti problemi di questo tipo.

Con l'aiuto delle idee viste fin qui, è possibile convertire moltissimi programmi per l'uso con i microdrive. Il tempo speso per il caricamento diventerà trascurabile, e ciò significa poter spendere più tempo utilizzando i programmi, e meno in attesa del loro caricamento.

Per rendere ancora più rapida la scelta dei vari programmi, può essere una buona idea realizzare un programma menu su ciascuna cartuccia, chiamato "Run", che elenchi tutti i programmi disponibili. Il programma 3 fornisce un esempio di programma menu, che riduce le operazioni per il caricamento di un qualsiasi programma presente sulla cartuccia attivandolo con la pressione di tre tasti: Run, Enter e n, dove n è il codice del programma scelto.

Listato 3 - Tipico file di menu per cartuccia microdrive.

```
1 REM *****
2 REM *
3 REM *      PROGRAMMA 3      *
4 REM *
5 REM *      esempio di file  *
6 REM *      "run"           *
7 REM *
8 REM *****
9 REM
10 BORDER 0: PAPER 0: INK 7
20 PRINT CHR$ 16;CHR$ 6;"MS
Vol 07 3/5/85"
30 PRINT
40 PRINT "1 - Fighter Pilot"
50 PRINT "2 - Manic Miner"
60 PRINT "3 - Jet Set Willy"
480 PRINT
490 PRINT CHR$ 16;CHR$ 6;"0 - U
scita"
500 IF INKEY$="1" THEN CLS : PR
```

```
INT AT 10,9;"Fighter Pilot": LOA
D *"m";1;"Fighter.c"CODE 30000:
RANDOMIZE USR 63792
510 IF INKEY$="2" THEN CLEAR 30
000: PRINT AT 10,10;"Manic Miner
": LOAD *"m";1;"Manic.c"CODE : P
RINT USR 33792
520 IF INKEY$="3" THEN CLEAR 30
000: LOAD *"m";1;"Jetset.c"CODE
: CLEAR 25000: PRINT USR 33792
980 IF INKEY$="0" THEN GO TO 10
000
990 GO TO 500
```

di Sergio Borsani

## Quali segreti ci riserva la stampa?

In questa puntata parlerò della stampante. So che molti non posseggono questo dispositivo, d'altra parte credo che una buona percentuale di coloro non sappiano sfruttarne le grandi capacità. Chi scrive dispone di una Epson MX-80 Type III, ma non fa alcuna differenza dal momento che il TI99/4A usa lo stesso modello rimarchiato Texas Instruments. Detto per inciso, lo stesso vale per l'IBM PC.

Le caratteristiche di stampa sono ben illustrate nel manuale, per cui cercherò di porre l'accento sui particolari meno evidenti piuttosto che su alcune possibilità alle quali non viene fatto alcun riferimento sullo stesso manuale della Texas, ma che sono presenti in quello originale della Epson.

Il computer comunica con la stampante tramite una porta seriale RS-232; pertanto ogni operazione richiede l'apertura di un canale con una Open. Diverso è il caso in cui si usi un'interfaccia parallela poiché in tal caso la scrittura o il listato di un programma si effettuano semplicemente con i comandi/istruzioni Lprint e Llist. Nel TI99/4A il trasferimento dei dati su stampante avviene previa istruzione Open # 1: "RS-232" o, per il listato, List "RS-232". Alla sigla RS-232, tra virgolette, dovranno seguire alcuni parametri a seconda di come sono predisposti i dip-switch interni della stampante. Se non vengono modificati i dip-switch sarà sufficiente la dicitura "RS-232". Tuttavia è molto frequente una diversa taratura della velocità di trasmissione dei dati e l'attivazione di otto fili di stampa.

La velocità di trasmissione è fissata dalla casa a 300 bps (bit al sec) e sono attivati solo 7 fili di stampa. Una diversa taratura richiede una esplicita dichiarazione nella Open. Ad esempio, Open # 1: "RS-232.BA = 4800.DA = 8" significa che la stampante è stata predisposta per ricevere dati alla velocità di 4.800 bps e che processa parole da otto bit (al posto di sette). Negli esempi e nei listati che seguiranno si userà l'istruzione Open # 1: "RS-232.DA = 8", ma è evidente che ognuno dovrà adattarla alle proprie esigenze. Chi non avesse attivato 8 fili di stampa non incapperà in alcuna situazione d'errore, ma non potrà sfruttare completamente le capacità grafiche della stampante. Buon per lui che qui si parla solo del modo testo.

Parliamo ora dei caratteri di controllo. La

stampa avviene in diverse modalità, con caratteri ingranditi, condensati, in grassetto, enfaticizzati e ancora con diversa larghezza del modulo oppure in modo testo o in modo grafico a singola e doppia densità, ecc. Tutto si ottiene inviando opportuni caratteri di controllo. Pensiamo che tutti sappiano che Print # 1: Chr\$(14); attiva la stampa in doppia larghezza e che Print # 1: Chr\$(15); attiva invece la stampa di caratteri condensati, 132 caratteri per riga. A volte si può restare imbarazzati di fronte a certe prestazioni. Personal Software, ed altre riviste, richiedono possibilmente listati su 40 colonne, per ovvie esigenze di impaginazione. Come ottenere la stampa su 40 colonne? È semplice, basta inviare i seguenti caratteri di controllo: Print # 1: Chr\$(27); Chr\$(81); Chr\$(40);.

### Listato 1 - Prova di stampa con caratteri normali, ingranditi, condensati, enfaticizzati e doppi.

```

100 OPEN #1: "RS232.DA=8"
110 PRINT #1: "CARATTERI NORMALI"
120 PRINT #1: CHR$(14);
130 PRINT #1: "CARATTERI INGRANDITI"
140 PRINT #1: CHR$(20); CHR$(15);
150 PRINT #1: "CARATTERI CONDENSATI"
160 PRINT #1: CHR$(18);
162 PRINT #1: CHR$(14); CHR$(15);
164 PRINT #1: "CARATTERI INGRANDITI E CONDENSATI"
166 PRINT #1: CHR$(20); CHR$(18);
170 PRINT #1: CHR$(27); CHR$(69);
180 PRINT #1: "CARATTERI ENFATICIZZATI"
190 PRINT #1: CHR$(27); CHR$(70);
200 PRINT #1: CHR$(27); CHR$(71);
210 PRINT #1: "CARATTERI DOPPI"
220 PRINT #1: CHR$(27); CHR$(72);
230 PRINT #1: CHR$(27); CHR$(69); CHR$(14);
240 PRINT #1: "CARATTERI INGRANDITI ED ENFATICIZZATI"
250 PRINT #1: CHR$(20);
260 PRINT #1: CHR$(27); CHR$(71);
270 PRINT #1: "CARATTERI ENFATICIZZATI E DOPPI"
280 PRINT #1: CHR$(27); CHR$(70);
290 PRINT #1: CHR$(27); CHR$(72);
300 CLOSE #1
310 END

```

CARATTERI NORMALI

CARATTERI INGRANDITI

CARATTERI CONDENSATI

CARATTERI INGRANDITI E CONDENSATI

CARATTERI ENFATICIZZATI

CARATTERI DOPPI

CARATTERI INGRANDITI ED ENFATICIZZATI

CARATTERI ENFATICIZZATI E DOPPI

# I SEGRETI DEI PERSONAL

Chr\$(81) si riferisce alla larghezza del modulo, Chr\$(40) al numero di colonne. In attesa di tali parametri il TI99 invia automaticamente un comando di ritorno carrello (CR) ogni 80 caratteri.

Qualcuno si sarà accorto che, pur specificando una larghezza di stampa di 132 colonne, ottenibile con caratteri condensati, si ha ugualmente un ritorno carrello dopo

## Listato 2 - Prova di stampa in compresso.

```
100 OPEN #1:"RS232.DA=8.CR"
110 PRINT #1:CHR$(15);CHR$(27);CHR$(81);CHR$(132);
120 FOR J=1 TO 6
130 PRINT #1:"STAMPA SU 132 COLONNE ";
140 NEXT J
150 PRINT #1:CHR$(13);CHR$(10);CHR$(18)
160 CLOSE #1
170 END
```

STAMPA SU 132 COLONNE STAMPA SU 132 COLONNE

80 colonne. La ragione sta in quanto detto in precedenza e per utilizzare tutta la larghezza del modulo bisogna inibire il CR automatico con: Open #1:"RS-232.DA=8.CR". Tuttavia, annullato il ritorno automatico, sarà necessario provvedere manualmente, per così dire, con i caratteri di controllo Chr\$(13); Chr\$(10), corrispondenti rispettivamente al ritorno carrello ed all'avanzamento riga (vedi listato 2.)

Un'altra caratteristica che riteniamo non venga molto usata, è quella che consente la stampa di dati sotto forma di tabelle. Lo schermo del TI99/4A è ahimé limitato con le sue 28 colonne di testo e volendo tabulare i dati elaborati ci si trova subito nei pasticci. In un programma pubblicato recentemente da **Personal Software**, *Ripartizione spese di riscaldamento*, il prospetto di ripartizione veniva scomposto sul video in

varie parti comprendenti due o al massimo tre colonne di dati. Il problema naturalmente non si pone sulla stampante, a patto di saper usare convenientemente i codici di tabulazione orizzontale. La stampante riconosce fino a 12 punti di arresto per la tabulazione ed ignora i successivi se ne vengono indicati in numero maggiore. I codici da inviare sono: Print #1:Chr\$(27);Chr\$(68);Chr\$(n1);Chr\$(n2);...;Chr\$(o), dove n1, n2, ... sono i punti di arresto espressi in numero di colonna. Per tabulare un dato si antepone ad esso il codice Chr\$(9); Prima di provare il listato 3 assicuratevi di annullare i codici dati in precedenza, relativi alla larghezza del modulo e alla stampa condensata. Il breve programma usa l'istruzione Print Using che si rende indispensabile qualora i dati siano in forma numerica; la corretta sintassi per l'output sulla stampante è: Print #1,Using "###.#":N;. Tra virgolette, dopo la parola chiave Using, c'è il formato di stampa che naturalmente può essere cambiato a seconda delle esigenze.

Ciò che manca assolutamente nel manuale della Texas Instruments è il riferimento alla sottolineatura e ai caratteri sovrascritti (esempio: esponenti nelle formule matematiche) e sottoscritti (esempio: indici). Per la sottolineatura bisogna inserire l'opzione CR nella Open per impedire il ritorno carrello automatico; dopo di che si può retrocedere il carrello, senza eseguire

## Listato 3 - Prova di tabulazione orizzontale.

```
100 OPEN #1:"RS232.DA=8"
110 DATA 345,54.8,22,15.3,999,456.3,728.4,9813
120 DATA 1.4,55.6,234,67.9,578.9,47,123.6,67
130 DATA 44,12.7,8365,87.3,87.1,599,834.6,6.2
140 PRINT #1:CHR$(27);CHR$(68);CHR$(10);CHR$(20);CHR$(30);CHR$(40);CHR$(50);CHR$(60);CHR$(70);CHR$(0);
150 FOR K=1 TO 3
160 READ N
170 PRINT #1,USING "#####.#":N;
180 FOR J=1 TO 7
190 READ N
200 PRINT #1:CHR$(9);
210 PRINT #1,USING "#####.#":N;
220 NEXT J
230 PRINT #1
240 NEXT K
250 CLOSE #1
260 END
```

345.0	54.8	22.0	15.3	999.0	456.3	728.4	9813.0
1.4	55.6	234.0	67.9	578.9	47.0	123.6	67.0
44.0	12.7	8365.0	87.3	87.1	599.0	834.6	6.2

## Listato 4 - Esempio di sottolineatura.

```

100 OPEN #1:"RS232.DA=8.CR"
110 PRINT #1:"TI-99/4A HOME COMPUTER";
120 PRINT #1:CHR$(13);
130 PRINT #1:RPT$(" ",22);
140 PRINT #1:CHR$(13);CHR$(10)
150 CLOSE #1
160 END

```

TI-99/4A HOME COMPUTER

l'avanzamento riga, con il codice Chr\$(13) e sovrastampare la lineetta per le sottolineature, (Fctn) (U). Il listato 4 riporta un breve esempio di sottolineatura.

Veniamo ora ai caratteri in sovrascrittura, intendendo per questa, la stampa di caratteri più piccoli di quelli normali, nella metà più alta della riga, come accade per gli esponenti nelle formule matematiche. Il codice da inviare è Esc S + (n), dove n può

assumere i valori 0 per la sovrascrittura e 1 per la sottoscrizione. In pratica bisogna scrivere: Print # 1:Chr\$(27); "S"; Chr\$(0); Chr\$(15);. Quando si inviano questi codici vengono stampati tutti i caratteri presenti nel buffer di stampa e quelli che seguono vengono stampati come caratteri sovrascritti nella metà superiore della riga. La stampa avviene in modo unidirezionale e in modo doppio. Dopo un primo passaggio la carta viene fatta avanzare di 1/216 di pollice e la testina passa una seconda volta. A causa di questo avanzamento la stampante procede automaticamente all'aggiustamento dell'interlinea per mantenere la lunghezza assoluta del modulo. Il codice

Esc T, cioè Chr\$(27);"T";, annulla la sovrascrittura, ma non la stampa doppia. Per questo è necessario al termine disattivarla con il codice Esc H.

Il listato 5 offre come esempio una equazione di terzo grado. Si ricorda che i codici "H", "T", "S", ecc, possono esser sostituiti con i rispettivi numeri di codice ASCII, Chr\$(72),Chr\$(84),Chr\$(83), ecc.

Non riteniamo con queste poche righe di aver esaurito un argomento che, come ben si sa, richiede un intero manuale; pensiamo tuttavia di aver toccato alcuni punti di un certo interesse che possono aprire la via ad un uso più ampio di questo indispensabile dispositivo che è la stampante, al di là del puro e semplice listato, alla quale viene spesso relegata.

Ci proponiamo, in un prossimo articolo, di sviluppare la parte che riguarda la stampa in Bit Image, cioè quella in modo grafico, alla quale, sia pur indirettamente abbiamo accennato nel numero di Marzo con il programma *Sub Copy* e che, come avrete modo di vedere, riserva ben altre sorprese. ■

## Listato 5 - Prova di stampa in grassetto con apici e pedici.

```

100 OPEN #1:"RS232.DA=8"
110 PRINT #1:CHR$(27); "E";
120 PRINT #1:"Y=aX";CHR$(27); "F";
130 PRINT #1:CHR$(27); "S";CHR$(0);
140 PRINT #1:CHR$(15); "3";
150 PRINT #1:CHR$(27); "T";CHR$(18);
160 PRINT #1:CHR$(27); "E";
170 PRINT #1:CHR$(27); "H";
180 PRINT #1:"+bX";CHR$(27); "F";
190 PRINT #1:CHR$(27); "S";CHR$(0);
200 PRINT #1:CHR$(15); "2";
210 PRINT #1:CHR$(27); "T";CHR$(18);
220 PRINT #1:CHR$(27); "E";
230 PRINT #1:CHR$(27); "H";
240 PRINT #1:"+cX+d"
250 CLOSE #1
260 END

```

$Y=aX^3+bX^2+cX+d$



# PERSONAL MARKET

## Apple

Vendo **Apple II europlus** in perfette condizioni a L. 600.000. Telefonare ore pasti. Ugo Rossini - Via Cadore, 31 - 20135 Milano - Tel. 02/4553588

Compro per Apple **IIc programmi gestionali, scientifici, didattici, linguaggi e giochi**, possibilmente con manuale. Inviare lista per contatto. Marco Chiesa - Via Gutuari, 5 - 14100 Asti - Tel. 0141/30605

Vendo per Apple **programmi per Totip-Enalotto-Totocalcio**. Sviluppo con selezione e stampa di qualsiasi sistema. Totocalcio già pronto per il 14?? risultato. Aldo Mena - Via Umberto I, 32 - 25030 Adro (BS) - Tel. 7356159

Cambio **programmi e giochi** per Apple IIe. Alberto Fasanotto - Via Campofiori, 55 - 37129 Verona - Tel. 045/23945

Cambio, vendo **listati vari** per Apple e Victor-Sirius. Salvo Giudice - Via San Pio X, 50 - 31031 Caerano San Marco (TV) - Tel. 0423/858201

Cambio **programmi di ogni tipo** per Apple II, IIe, IIc e compatibili. Per maggiori informazioni scrivete o mandate la vostra lista, risponderò con la mia. Igor Bonat - Via Resson, 6 - 34141 Trieste - Tel. 040/768718

Cambio per Apple **ogni tipo di software**. Per informazioni scrivere a: Giacomo Cracchiolo - Piazza M. Venturi, 30 - 66100 Chieti - Tel. 0871/66687

Cambio **software per Apple II con hardware di ogni tipo**. Ho una vastissima soffeca con programmi per ogni esigenza. Gian Paolo Capponi - Via Riva del Garda, 98 - 41100 Modena - Tel. 059/302926

Vendo per Apple II, avendo cambiato sistema, alcuni programmi fra cui: **Multipan, Pfs-File, UCDS-Pascal, Paddlepaint, Spreadsheet, CP/M e ProDOS** a prezzi stracciati. Telefonare a: Nicola - Tel. 0362/522476

Vendo **Apple IIe più monitor II, drive**, 50 programmi tra cui Visidex, VisiCalc, Flight Simulator, Avventura Nel Castello, ecc. Il Tutto, in perfette condizioni, a L. 2.000.000. Giampaolo Rolli - Via Rodolfo Tanzi, 41 - 43100 Parma - Tel. 0521/31303

Vendo **Apple II plus originale** a L. 790.000; separatamente anche drive originale a L. 460.000; scheda language 16 Kbyte a L. 100.000; scheda CP/M nuova a L. 125.000; interfaccia parallela grafica per stampante a L. 110.000; **scheda 80 colonne più 64 Kbyte per IIe** a L. 210.000. Telefonare ore serali. Roberto Rossi - Via Lario, 26 - 20159 Milano - Tel. 02/6070236

Cerco **paddles o joystick** per Apple II plus solo se in buono stato di funzionamento. Telefonare ore serali. Paolo Bonato - Piazza Tirana, 6/E - 20147 Milano - Tel. 02/4152866

Vendo per Apple II **"Supertoto 1.0" superprogramma Totocalcio inedito**, 3 diverse opzioni di selezioni incrociate (nr. segni 1X2; consecutivi; corr. errori), con output nr. colonne utili, sviluppo su monitor o stampante a L. 60.000 più manuale. Roberto Rossi - Via Lario, 26 - 20159 Milano - Tel. 02/607023

## Commodore

Vendo **VIC 20 più joystick**, "Introduzione al BASIC" parti 1 e 2, 3 cartridge, registratore a L. 300.000. Roberto Carotti - Via Benedetto Croce, 142 - 52100 Arezzo - Tel. 0575/352982

Vendo **VIC 20 più registratore**, joystick, 3 cartucce, molti programmi in cassetta, 2 libri sul VIC 20, moltissimi listati. Antonio Pavan - Via Jacopo Della Quercia, 48 - 35134 Padova - Tel. 049/612447

Compro e cambio **manuali di programmi per C64**, massima serietà. Cerco amici interessati al FORTH, sono interessato a qualsiasi cosa (programmi, libri, riviste) sul FORTH. Ferruccio Cantone - Via Genova, 18 - 10076 Nole (TO) - Tel. 9297125

Vendo **C64, registratore, riviste** insieme a 3 cartridge al miglior offerente. Prezzo base: L. 290.000. Telefonate! Roberto Bruccoleri - Via Torino, 58 - 12044 Centallo (CN) - Tel. 0171/714704

■ Cerco in **Vicenza e provincia appassionati/e di C64** per scambio di programmi ed esperienze per realizzare iniziative comuni come l'acquisto di software a spese ripartite e club. Andrea Lorenzon - Via Luigi Cavalli, 5 - 36100 Vicenza - Tel. 0444/40050

Vendo **Commodore 64 più drive 1541**, registratore originale, 2 dischi con programmi, 10 cassette per un totale di circa 500 programmi. Il tutto a L. 1.400.000 trattabili. Vendo anche i pezzi separatamente. Enrico Minelle - Corso Vittorio Emanuele, 165 - 35100 Padova - Tel. 049/663387

■ Cerchiamo **adesioni** per costituzione del più favoloso dei Commodore Club. Telefonare ore pasti. Davide Conte e Donato Aprile - Via Antonio De Luca, 10/G - 80070 Ischia (NA) - Tel. 081/99108

Vendesi **Commodore C16 seminuovo con registratore 1531**, 14 giochi originali su cassetta (Donkey Kong, Harbour Attack ...), joystick Magnum con autofire a L. 400.000 trattabili. Maurizio Cima - Via Giuseppe Verdi, 11 - 55048 Torre del Lago (LU) - Tel. 0584/342024

Vendo **VIC 20 più 16 Kbyte, registratore originale**, centinaia di programmi, trislot autocostituito a L. 250.000. Telefonare ora di cena. Alessandro Giolitti - Via Fabroni, 45 - 50134 Firenze - Tel. 055/473810

Vendo **VIC 20 più espansione 16 Kbyte**, super expander, cartuccia machine code language, cartucce di giochi e nastro di programmi a L. 350.000. Riccardo Bozzi - Via Risorgimento, 90 - 20092 Cinisello Balsamo (MI) - Tel. 6128851

Compro per C64 **schede aggiuntive di ogni tipo**. Ferruccio Cantone - Via Genova, 18 - 10076 Nole (TO) - Tel. 011/9297125

Cambio **più di 1.500 programmi** per C64. Rispondo solo a chi vuole scambiare. Scrivi anche se non possiedi molti programmi, potresti avere quello che fa per me. Bruno Castelnovo - Via S. Antonio, 3 - 29020 Montagna in Valtellina (SO) - Tel. 0342/380234

Vendo **Commodore 64 più floppy disk**, registratore a nastro, stampante, vari testi, linguaggi, diversi programmi gestionali e giochi. Telefonare ore ufficio. Mauri Renato - Via Manzoni, 1 - 20080 Zibido S. Giacomo (MI) - Tel. 9003105

Cambio **programmi per C64, C16 e Plus/4**. Dispongo di numerosi manuali. Cerco anche routines grafiche per MPS-802. Richiedere liste a: Gerardo Marra - Via P. Baratta, 91/A - 84091 Battipaglia (SA) - Tel. 0828/23130

Cerco **programma Fortran** per CBM 64. Cambio con programma a scelta tra The Last One, Logo ed altri. Telefonare ore pasti (solo zona Milano). Luca Candaten - Via Senigallia, 11 - 20161 Milano - Tel. 02/6468851

Vendo **espansione 16 Kbyte e registratore dedicato** per VIC 20. Telefonare a: Danilo Bartoli - Via Amendola, 7 - 25086 Rezzato (BS) - Tel. 2791586 (ore 19.00 - 20.00)

Attenzione! Causa passaggio a CBM 64, vendo **VIC 20 più registratore**, libro "Alla scoperta del VIC 20", due cassette gioco a sole L. 200.000 trattabili. Tutto in ottimo stato (comprato nel Gennaio 1985). Contatto tutta Italia. Per informazioni scrivere a: Gennaro Pinto - Via Seripando, 28 - 64100 Salerno

Compro a L. 500.000 **Commodore 64 più disk drive**, registratore dedicato; plotter 1520 a L. 150.000. Vendo inoltre Atari 600XL più registratore a L. 299.000. Massimo Iacomelli - Via Piranesi, 26 - 20 137 Milano - Tel. 02/7490510

Vendo al miglior offerente **Commodore CBM 4032 più dual drive 4040**, il tutto in perfette condizioni e con molti programmi. Dott. Lucio Crippa - Via F. Nullo, 5 - 20035 Lissone (MI) - Tel. 039/481485

Cambio **30 videogames per C64** (One On One, Fort Apocalypse, Wimbledon, Mundial Soccer, ecc.) per Tot 13 o Magic Desk più Easy Script. Per contatti scrivere o telefonare. Salvatore Argonizza - Via S. Francesco, 45 - 87060 S. Giorgio Albanese (CS) - Tel. 0983/86112

Vendo **CBM 64 più floppy, datassette**, monitor colore, 4 libri, tutto il miglior software esistente su dischetti; tutto come nuovo, usato pochissimo, disponibile per dimostrazioni a L. 1.600.000. Luigi Callegari - Via De Gasperi, 47 - 21040 Sumirago (VA) - Tel. 0331/909183

Vendo per Commodore 64 **programma di contabilità ordinaria**: prima nota, libro giornale, schede clienti/fornitori e conti di reddito; stampa bilancio e situazione clienti/fornitori, il tutto automaticamente. Rag. Luciano Uselli - Via Isonzo, 31 - 21100 Varese - Tel. 0332/242596

Vendo **VIC 20, datassette, 30 giochi**, alcune riviste e libri, tutto in ottime condizioni (esiste ancora la garanzia). Tutto a L. 180.000. Marilena Fontanesi - Via C. Tenca, 5 - 20124 Milano - Tel. 02/6594598

Si è aperto in Campania un club riservato ai soli utilizzatori di C64.

L'iscrizione è gratuita. Tutti gli iscritti riceveranno ogni mese un bollettino ricco di programmi e di trucchi. Ezio Esposito - Via Bonea, 1 - 80069 Vico Equense (NA) - Tel. 081/8798584

Vendo per VIC 20 **espansione 8 Kbyte RAM, cassetta game "Sirpent" originale**, cassetta con 5 game più libro "Alla scoperta del VIC 20" e libro con 20 programmi. Il tutto a sole L. 100.000. Stefano Bonelli - Via G. Di Vittorio, 7 - 53014 Monteroni D'Arabia (SI)

Per CBM 64 dispongo di **libri, manuali, utility su cartridge** (Turbo, Fast Copy, Turbo Disco, ecc.). Marcello Cesi - Via Magliana Nuova, 178 - 00146 Roma - Tel. 06/5266009

Vendo **VIC 20 (1° anno di vita) più espansione 16 Kbyte** modificata per duplicare cartucce su nastro, 50 programmi con utility e games, super expander a L. 250.000. Tratto solo in Calabria. Angelo Lacco - Via Degli Stadi, 80 - 87100 Cosenza - Tel. 0984/31871

Vendo **C64 più floppy, datassette**, 30 dischi, libri dedicati a L. 1.100.000. Il software è una selezione del meglio presente sul mercato. Materiale in perfette condizioni. Luigi Callegari - Via De Gasperi, 47 - 21040 Sumirago (VA) - Tel. 0331/90918

Vendo **Commodore 64 più registratore**, drive 1541, stampante MPS802, 180 magnifici programmi, manuali vari, tutto come nuovo con imballi originali a L. 1.600.000 o separatamente. Massimo Proia - Via Pubblico Passeggio, 16 - 29100 Piacenza (PC) - Tel. 0523/32417

Vendo **VIC 20 accessoriatissimo**: Eprom, Motherboard, 3 Kbyte S.E., registratore, light pen, libri riviste, software LM; il tutto a sole L. 450.000 trattabili. Telefonare ore pasti. Massima serietà. Silvio Cappelli - Via Milano, 19 - 19100 La Spezia - Tel. 0187/36600

Vendo **VIC 20 più espansione 16 Kbyte** più 3 cartucce (VIC-Avenger, Dragonfire, Sargon II Chess), tante cassette; il tutto a L. 180.000. Rivolgerti ore serali a: Marco Della Sala - Via Francesco Ferraironi, 41/7 - 00172 Roma - Tel. 06/288472

Sono da poco un possessore di CBM 64 e sono molto interessato all'acquisto di programmi su nastro: utility, giochi e programmi vari. Prego inviarmi vostre liste. Pasquale Martire - Via Emanuele Carnevale, 1 - 90145 Palermo - Tel. 091/563931

Vendo **C64 (10 mesi di vita) più drive**, registratore dedicato, 50 utility, 100 giochi LM. Chiamatemi, troveremo un accordo. Benedetto Raspanti - Via Dei Glicini, 37 - 91100 Trapani - Tel. 0924/22911

Compro **disk driver 1541** per CBM 64 ad un buon prezzo e in buono stato con software (solo zona Bologna). Stefano Pierotti - Piazza Musi, 11 - 40127 Bologna - Tel. 051/500174

Vendo **VIC 20** a L. 200.000 trattabili. Per chi acquista il VIC 20 libri e cassette in omaggio. Massimo Vannetti - Via Paolini, 7 - 58043 Castiglione della Pescaia (GR) - Tel. 933841

Compro **C64 usato purchè in buono stato** e vera occasione più registratore, stampante MPS802. Acquisto anche singolarmente. Tratto preferibilmente con le zone di Puglia e Basilicata. Maurizio Coccorese - Via Gentile, 108/D - 70126 Bari - Tel. 080/491374

**Riparo Commodore 64 e Spectrum** in tempi brevi. Telefonare ore ufficio. Carlo Merlini - Via Ravenna, 14/A - 20135 Milano - Tel. 02/5695876

Vendo **VIC 20 più registratore C2N**, Super Expander, Hes Mon, joystick, cartucce (Scacchi, Golf), giochi su cassetta, 4 libri, il tutto a L. 300.000. Roberto Stanghellini - Via Italia, 40 - 20094 Corsico - Tel. 02/4583257

Vendo **plotter Commodore 1520 usato pochissimo** solo in zona Roma a L. 290.000 non trattabili. Telefonare ore pasti a: Alessandro Fabbrini - Via Ludovico di Savoia, 10 - 00185 Roma - Tel. 06/7578609

Per C64 vendo **cartridge Turtle Graphics II** a L. 70.000 usata pochissimo. Telefonare ore 19.00-21.00 a: Lello Bove - Via Papini, 29 - 80046 San Giorgio a Cremano (NA) - Tel. 081/7714412

Vendo **Commodore VIC 20 in buono stato più registratore**, joystick con molti giochi ed espansione 16 Kbyte. Il tutto a sole L. 300.000 trattabili. Marco Mori - Via A. Vespucci, 7 - 50054 Fucecchio (FI) - Tel. 0571/261088

**Commodore Club Perugia cerca soci in tutta Italia**. Iscrizione L. 3.000 mensili; videogame, scambi e tante iniziative. Simone Merlini - Via Baglioni, 24 - 06100 Perugia - Tel. 075/61580

Vendo **C64 più drive 1541 e regalo software all'acquirente**. Tutto funziona perfettamente! Il prezzo? Solo L. 800.000! Telefonate a: Paolo Toffetti - Via Derna, 28 - 20132 Milano - Tel. 02/2829186

Vendo **VIC 20 più relativi cavetti, espansione 8 Kbyte**, cassette con circa 300 giochi in LM, 3 libri (VIC Programmers, Alla scoperta del VIC, Impariamo a programmare) più 4 cartucce a L. 200.000. Roberto Zannoni - Via Biancardi, 9 - 20075 Lodi (MI) - Tel. 0371/53847

Per C64 cambio **qualsiasi programma**, sia games che utilità o gestionali su disco. Inviatemi la vostra lista, assicuro massima serietà e risposta a tutti. Umberto Brunetti - Via Rosa Luxemburg, 21 - 47040 Ospedaletto di Rimini (FO)

Vendo **VIC 20 più registratore**, super expander 3 Kbyte, Cosmic Cruncher (su cartuccia), 50 stupendi giochi tra cui Tron, Defender, Pac Man, Arcadia, Amok, ecc. Il tutto a sole L. 300.000 trattabili. Fabio Boiardi - Via Sidoli, 50 - 29100 Piacenza - Tel. 0523/72817

Vendo **per VIC 20**: 16 Kbyte RAM L. 75.000, 8 Kbyte RAM L. 45.000, 24 Kbyte RAM L. 110.000, VIC Mon L. 35.000, Hi-Res L. 35.000, programmatore di 2516/2716 L. 30.000. Per C64 vendo a L. 20.000 disco con utility (copiatori, compilatori). Nicola Pedroli - Casella Postale - 20080 Cisliano (MI) - Tel. 02/9018875

Vendo **C64 più registratore, copritastiera**, garanzia Sirius, joystick, tastierino musicale, oltre 40 cassette, 8 libri eccezionali e costosissimi; il tutto a L. 700.000. L'intero sistema ha solo 2 mesi di vita. Fabrizio Benelli - Via Don Pio Vannucchi, 41/A - 50047 Prato (FI) - Tel. 0574/594942

Vendo **Commodore CBM 3032 a fosfori verdi 32 Kbyte RAM**, con registratore C2N, manuali, a L. 500.000 trattabili. Telefonare chiedendo di Herbert o scrivere. Carlo Puccio - Via Litta Modignani, 5 - 20161 Milano - Tel. 02/6464454

## Sinclair

Cerco **Sinclair ZX80 con vecchia o nuova ROM** e schema elettrico dello ZX80. Telefonare solo tra le 12.30 e le 13.00 o tra le 19.30 e 20.30. Aldo Vendramin - Via Chiaradia, 3 - 33077 Sacile (PN) - Tel. 0434/733139

Disponendo di 1.200 programmi per ZX Spectrum **cerco utenti disposti allo scambio Spectrum - Commodore 64**. Annuncio sempre valido. Alfredo Trifiletti - Via Fiume, 20/A - 71100 Foggia - Tel. 0881/75385

Vendo **ZX Spectrum 48 Kbyte più 100 programmi**, interfaccia joystick programmabile a L. 350.000 trattabili. Inoltre vendo programmi per Apple IIe - II plus. Stefano Sparvoli - Via Passo Rolle, 53 - 20134 Milano - Tel. 2151496

Causa passaggio a sistema superiore vendo agli "apprendisti" di BASIC **Sinclair ZX81 più alimentatore**, manuale, espansione 16 Kbyte, programmi su cassetta a sole L. 70.000! Scrivere o telefonare. Antonello Carbone - Via Adone, 5 - 95040 S.G. Galermo (CT) - Tel. 095/393046

Vendo **Spectrum 48 Kbyte** a L. 300.000; 350 programmi L. 150.000; interfaccia joystick L. 50.000; registratore L. 50.000. In blocco o separatamente; telefonare ore pasti, solo zona Roma. Giuseppe Fasulo - Via F. Berni, 5 - 00185 Roma - Tel. 7577351

Vendo **Spectrum 48 Kbyte con manuali**, registratore, libro per LM, 2 libri di grafica, light pen, cassette giochi, giornali. Acquistato nel Dicembre 1984, tutto in ottime condizioni a L. 400.000. Mauro Pavone - Via Capececelaro, 30 - 20148 Milano - Tel. 02/4075792

Vendo **Sinclair ZX80** a L. 100.000. Andrea Roidi - Viale Augusto, 105 - 80125 Napoli - Tel. 081/616837

**Quantum Leap User Club cerca nuovi soci in tutta Italia**. Iscrizione gratuita. Abbiamo già a disposizione libri, software e tanti consigli e idee. Roberto Ghezzi - Via Volontari Del Sangue, 202 - 20099 Sesto S. Giovanni (MI) - Tel. 02/2485511

Desidero comunicare con possessori di Sinclair QL per **scambio informazioni, chiarimenti, ecc.** Ulderico Guadagno - Via Tito Livio, 95 - 00136 Roma - Tel. 06/3492694

Vendo **ZX Spectrum 48 Kbyte più alimentatore stabilizzato**, stampante Alphacom 32, numerosi programmi, tutto a L. 450.000. Telefonare ore pasti. Giuseppe Pingitore - Corso G. Nicotera, 137 - 88046 Lamezia Terme (CZ) - Tel. 0968/23816

Compro, cambio **software per QL**. Richiedere e/o inviare lista. Andrea Galli - Via Palagetta, 212 - 50017 S. Piero a Ponti (FI)

Vendo **ZX Spectrum 48 Kbyte Issue 3 (6 mesi di vita)** con alimentatore anti black-out e predisposto per l'attacco monitor a L. 320.000. Giorgio Colombo - Via S. Carlo, 13 - 20035 Lissone (MI) - Tel. 039/481908

Compro **ZX Interface 1 più ZX microdrive con cartucce**, possibilmente in ottimo stato. Compro anche separatamente. Prezzo contenuto. Gianni Remo - Via Bandita, 20/D - 51022 Bardalione (PT) - Tel. 65318

# PERSONAL MARKET

Vendo **ZX81 più espansione 64 Kbyte**, alimentatore, cavetti, 3 libri (manuale originale, "Guida allo ZX81", "66 programmi per ZX81"). Imperio Palazzini - Via Pisino, 36 - 00177 Roma - Tel. 06/2593893

Vendo per Spectrum 48 Kbyte **programma di riduzione sistemi Totocalcio in LM**, con possibilità di stampare le schedine con Seiksha GP 50S, a L. 30.000. Per informazioni telefonare a: Claudio Nava - Via della Palazzetta, 9 - 20052 Monza (MI) - Tel. 039/742638

Il Sinclair Club di Roma-Talenti **cerca possessori di Spectrum in tutta Italia** tra coloro che vogliono usare il computer sia come sofisticato videogioco o cerchino qualcosa in più. Massimo D'Ascenzo - Via F. D'Ovidio, 109 - 00137 Roma - Tel. 06/8280043

Vendo **ZX Spectrum più espansione** a L. 300.000 trattabili con omaggio di 100 giochi, 16/48 Kbyte. Telefonare dopo pranzo o scrivere a: Daniele Cinquini - Via Volpi, 39 - 55049 Viareggio (LU) - Tel. 391196

Vendo **ZX Spectrum 80 Kbyte** a L. 300.000, compresi programmi con le migliori utility e i più bei videogiochi. Tutto come nuovo. Marco Prinetti - Via Mazzone, 14 - 13037 Serravalle Sesia (VC) - Tel. 0163/470177

Cambio **programmi di ogni tipo** per ZX Spectrum. Per maggiori informazioni scrivere e mandare la propria lista, risponderò con la mia. Igor Bonat - Via Ressa, 6 - 34141 Trieste - Tel. 040/768718

Vendo **Spectrum 48 Kbyte, registratore in omaggio**, libro sullo Spectrum, 6 cassette di giochi; tutto a L. 300.000 trattabili. Il prezzo comprende spese di spedizione. Simone Dall'Oglio - Via Caduti sul Lavoro, 4 - 52100 Arezzo - Tel. 0575/24729

Vendo **Spectrum 48 Kbyte completo di accessori**, manuali, joystick interfacciato, 250 programmi a L. 450.000 trattabili. Marcello Tontodonati - Piazza Vescovo, 3 - 00199 Roma - Tel. 06/8313266

Vendo X Spectrum, alimentatore, cavi, manuale in italiano, interfaccia, joystick, 100 giochi, riviste. Il computer è espanso a 48 Kbyte. Tutto in perfette condizioni a L. 350.000. Sergio Ressi - Via Bernardino Verro, 78/F - 20141 Milano - Tel. 8466551

Vendo **ZX Spectrum più 2 manuali in italiano originali** ed oltre 200 programmi a L. 300.000 trattabili. Telefonare dopo le 19.00. Massimiliano Poletti - Via Val d'Intelvi, 4 - 20152 Milano - Tel. 02/4590357

Affarone! Vendo **ZX Spectrum come nuovo con manuale in italiano e inglese, ZX printer, registratore, listati**, libro sul linguaggio macchina. Il tutto a L. 400.000. Telefonare ore pasti. Roberto Tirone - Via Pacini, 76 - 20131 Milano - Tel. 02/295064

Per Sinclair QL vendesi **espansione 128 Kbyte RAM**. Telefonare per informazioni ed accordi. Giampiero Sobrero - Via Apparizione, 15 - 16133 Genova - Tel. 010/392187

Vendo **ZX Spectrum più interfaccia joystick Kempston programmabile** (adatta a qualsiasi gioco) con istruzioni in italiano, il libro "Alla scoperta dello ZX Spectrum", circa 200 giochi tra i quali le ultime novità come Ghostbusters, Pole Position, Beach Head, ecc., diverse riviste italiane e inglesi. Tutto a L. 380.000 trattabili. Gianluca Papi - Via Teognide, 134 - 00125 Roma - Tel. 06/605485

## Texas

Vendo **modulo Extended BASIC più manuali**, corso su nastro a L. 150.000; consolle TI 99/4A, manuali, accessori, joystick Spectravideo adattato a L. 200.000. Solo zona Roma - Telefonare ore pasti. Claudio Fedeli - Piazza F.A. Doria Pamphili, 23 - 00149 Roma - Tel. 06/5584418

Vendo per passaggio a sistema superiore **TI 99/4A più cavo per registratore**, registratore, manuale in italiano, 2 libri di programmi, 3 cassette giochi, listati a L. 280.000; tutto in condizioni ottime, solo due mesi di vita. Claudio Sala - Via Avanzini, 12 - 41042 Fiorano (MO) - Tel. 0536/830451

Vendo al miglior offerente **TI 99/4A completo di espansione memoria 32 Kbyte**, interfaccia parallela, RS232, modulo Extended BASIC, TiCalc, Scacchi, Calcio, Invaders, Carwars, Munchman, Videogiochi 1, Parsec, Moon Mine, joystick, registratore, 150 programmi su cassetta, 1.000 listati, 5 volumi. Telefonare ore pasti serali. Gigi Masala - Via G. Casati, 47 - 20033 Palazzolo Milanese (MI) - Tel. 02/9101693

Compro **Extended BASIC** a prezzo stracciato con testo Texas Instruments TI 99/4A. Marco Pallagrosi - Via Tor Cervara 110, 112 - 00151 Roma - Tel. 06/221896

Vendo per TI 99/4A **programma su cassetta di "Gestione Finanziaria"** con calcolo rendite, sconti, interessi composti e tabelle ammortamento costante e progressivo (util. maschere video) a L. 40.000. Marco Pedroni - Via Mazzini, 22 - 26100 Cremona - Tel. 0372/37265

Compro per TI 99/4A **modulo SSS Soccer** ad un prezzo trattabile ed inoltre cambio un sacco di giochi, veramente belli, sia su nastro che listato. Scrivetemi presto e a tutta forza! Claudio Fameli - Via Leopoldo Bignone, 27/5 - 16157 Ge-Pra (GE) - Tel. 010/663998

Compro per TI 99/4A **coppia di joystick**. Inoltre possiedo un TI 99/4A non funzionante. Cerco laboratorio per eventuale riparazione. Chi può fornirmi indirizzi utili mi scriva. Luigi Garofoli - Via Ospedale, 29 - 60011 Arcevia (AN)

Per TI 99/4A compro **interfaccia RS232 perfettamente funzionante**. Scrivere o telefonare ore serali. Alessandro Garoli - Corso Dante, 35 - 12100 Cuneo - Tel. 65265

Compro **manuale per Extended BASIC scritto in italiano**. Massimiliano Bruno Ventre - Via De Galboli, 7 - 28100 Novara - Tel. 0321/478356

Vendo per TI 99/4A **programma su cassetta per calcolo "attenuatori" a "T", "T-shuntato" e "TT" con rappresentazione grafica**, a L. 35.000. Scrivere o telefonare ore pasti. Marco Pedroni - Via Mazzini, 22 - 26100 Cremona - Tel. 0372/37265

Compro **modulo SSS Minimemory** per TI 99/4A. Prezzo massimo L. 90.000. Inviare offerte a: Maurizio Ortolani - Viale Dante Alighieri, 41 - 61100 Pesaro

Vendo **TI 99/4A, joystick, Extended BASIC**, 10 moduli SSS con istruzioni, 10 cassette di giochi vari, manuali e listing diversi a L. 550.000. Luca Cenacchi - Via Sabotino, 4 - 21047 Saronno (VA) - Tel. 02/9607196

Vendo **TI 99/4A in ottime condizioni** completo di cavi TV e registratore, modulo SSS Scacchi, manuale di istruzioni, libro sul TI 99 del Gruppo Editoriale Jackson, cassette di programmi. Davide Pignat - Pendice Scoglietto, 5/3 - 34127 Trieste - Tel. 0432/577275

Vendo **TI 99/4A più cassetto per registratore**, 2 manuali, 30 programmi su cassetta (Jawbreaker II, Moon Mine, Archivio, ecc.), programmi su carta, tutto nuovissimo (3 mesi di vita) a L. 340.000. Stefano Rostici - Via della Selva, 81 - 00034 Colleferro (RM) - Tel. 9780280

Compro **peripheral box, completo di sistema a dischi** a L. 500.000 trattabili, tutto completo dei relativi manuali. Maurizio Ortolani - Viale Dante Alighieri, 41 - 61100 Pesaro

Vendo **TI 99/4A più alimentatore, modulatore PAL**, cavo per registratore, 11 programmi su cassetta, libro sul TI e "Giochiamo col TI", riviste "MC Microcomputer" e 20 giochi su fotocopia a L. 200.000 trattabili. Maurizio Bua - Via Marcello Capra, 57 - 94014 Nicosia (EN) - Tel. 0935/48179

Cerco **P-Code completo di accessori** in buono stato. Telefonare ore pasti. Leonardo De Miccolis - S.S. Alberobello, 80 - 70017 Putignano (BA) - Tel. 080/734683

Vendo **TI 99/4A più expansion box, Extended BASIC**, Editor Assembler, 32 Kbyte RAM expansion, disk drive, 3 moduli SSS. Regalo dischetti, cassette, joystick e cassetto registratore. Tutto con manuali originali più Disk Manager. Vincenzo Vidili - Corso Grosseto, 62 - 10148 Torino - Tel. 011/2201809

Vendo **TI 99/4A completo con coppia joystick** e diversi programmi, più SSS Filetto a L. 200.000. Giuseppe Bove - Via R. Follereau, 45 - 60044 Fabriano (AN) - Tel. 261814

Vendo **TI 99/4A con Extended BASIC**, 3 moduli SSS, 2 joystick, cavo per registratore, programmi su cassetta, registratore Texas dedicato, manuali, corsi, libri. Telefonare ore 19.00-21.00. Hubert Schincariol - Via Vittorio Veneto, 104 - 22032 Albese con Cassano (CO) - Tel. 031/42764

## Varie

Sviluppiamo o personalizziamo software su richiesta su **CBM, Apple e MS-DOS**. Preventivi su richiesta telefonica. Serietà e competenza. Per informazioni e richieste scrivere o telefonare. Alfredo Dal Ferro - Via Fornare, 3 - 21040 Castronno (VA) - Tel. 0332/495989

Vendo **PB700 Casio completo di plotter e registratore** CN1 con valigetta e libro, interfaccia parallela per stampante FA4, libreria con numerosi programmi. Tutto a L. 1.100.000. Armando Scudieri - Via Aurelia Pon., 50 - 18011 Arma di Taggia (IM) - Tel. 0184/83284

Vendo **Sharp MZ-700 più stampante, plotter 4 colori**, registratore, BASIC, manuale. Usato poco, ottime condizioni, L. 800.000 non trattabili. Vincenzo Cusolito - Via Setaioli, 15 - 98100 Messina - Tel. 090/362459

Vendesi **PC 1500 Sharp più stampante**, espansione 8 Kbyte, adattatore a L. 700.000 trattabili. Alfio Masier - Via Balustra - 15057 Tortona (AL) - Tel. 0131/812434

Cerco rivista arretrata di "Elektor" nr. 13 (Giugno 1980) per acquistarla o, eventualmente, fotocopiarla. Per proposte scrivere oppure telefonare ore pasti. Pago bene anche solo per fotocopiarla. Maurizio Speciale - Via Merano, 3 - 20093 Cologno Monzese (MI) - Tel. 02/2543509

Cerco **fotocopie manuale del compilatore BASIC-IBM "BASIC Compiler" by Microsoft** in inglese/italiano. Telefonare ore ufficio allo 0832/26482 chiedendo di Sergio. Sergio Nuzzo - Via Q. Sella, 3 - 73100 Lecce (LE)

Vendo **Triumph Adler PC con floppy e monitor 12" fosfori verdi, 3 mesi di vita**, con CP/M, BASIC interprete e compilatore, Assembler, MBASIC, WordStar, LogiCalc e software vario a L. 2.000.000. Telefonare ore ufficio a Roberto, tel. 02/2899763

Vendo **computer portatile Olivetti M10 24Kbyte con parecchi programmi** (traduttore Pascal, compositore musicale, studio di funzioni). Ideale per studenti (io lo porto agli esami al Politecnico), L. 1.500.000 trattabili. Guido Galimberti - Via F.lli Rosselli, 9 - 20139 Milano - Tel. 02/568660

Cerco **fotocopye dei listati e relativi articoli** di Abukir 1798 ("Personal Software" nr. 16 - 1984), Le Mans ("Bit" nr. 15 - 1984). In cambio di ogni programma inviatomi offro 2 giochi per CBM 64. Gianluca Ferri - Via Roma, 16/1 - 48018 Faenza (RA) - Tel. 0546/662263

Vendo **Sega SC-3000 più cartuccia memoria 16 e 32 Kbyte**, cassetta "Impariamo il BASIC", joystick, 3 cartidge gioco (Insub, Golf, Pacar) a L. 500.000. Guido Pedrini - Via Villini di Sturla, 2/3 - 16131 Genova - Tel. 010/3994223

Vendo Atari CX 2600 in buono stato; con 6 cassette gioco L. 350.000, senza cassette L. 200.000. Scrivere a: Silvio Peruzzo - Via Vignole, 1 - 36030 Levà di Montecchio P. (VI) - Tel. 0445/864616

Vendo computer Spectravideo SV318 seminuovo, 3 mesi di vita (64 Kbyte ROM/RAM), registratore, alimentatore, cavo TV, 3 cassette di giochi, cartridge gioco Sectoralpha, manuali d'uso a L. 420.000 (trattabili). Alex Fantechi - Via Bello, 16 - 17030 S. Fedele (Albenga) (SV) - Tel. 548256

Vendo Intellevision più Intellivoice, Lucky Keyboard, 10 cassette (Burgertime, River Raid ...), istruzioni e scatole originali a L. 650.000 trattabili. Telefonare ore serali. Danila Ronco - Corso Vittorio Emanuele II, 200 - 10138 Torino - Tel. 011/7567118

Vendo MPF II 64 Kbyte RAM, Applesoft compatibile più tastiera separata, interfaccia per 2 drive, 1 drive, joystick, 10 cassette giochi, 10 dischi utilità, tutti i manuali italiani a L. 1.000.000 (tratt.). Giuseppe Culmone - Via 2/F, 11 - 91011 Alcamo (TP) - Tel. 0924/29158

Vendo box, disk drive, disk controller, disk manager, il tutto garantito, per L. 1.500.000 non trattabili. Solo zone di Milano, Bergamo, Lecco. Telefonare di domenica. Carlo Aceti - Via I Maggio, 4 - 92050 Verderio Inferiore (CO) - Tel. 039/510715

Vendo Sega SC-3000 16 Kbyte espandibili, alimentatore stabilizzato, cavetti, manuale d'istruzioni, cartuccia BASIC, Star Jaker, Congo Bongo, 2 joystick semiprofessionali a L. 350.000 (valore effettivo L. 750.000). Marco Moniga - Via Collecchio, 16/8 - 20148 Milano - Tel. 02/395140

Vendo console Atari 2600 a L. 150.000 più 22 cassette tra le più belle in commercio a L. 500.000. Vendo tutto in blocco a L. 600.000. Milo Gallico - Via Boeri, 11 - 20141 Milano - Tel. 02/8431317

Vendo stampante Epson FX80 con interfaccia parallela Apple a L. 1.200.000 fatturabili. Telefonare ore pasti. Raul Rosenthal - Via Montebello, 67 - 25128 Brescia - Tel. 030/380946

Vendo a prezzo da concordare Olivetti M20-ST doppio floppy, video fosfori verdi, stampante PR 1450; software ingegneria civile, gestione ambulatorio U.S.L. di medicina generale. Oliword, Ollitutor, Gestione Archivi Clienti, vari. Telefonare ore serali. Dott. Francesco Ferrante - Via I Maggio, 33 - 56030 Montecatini (PI) - Tel. 0587/748457 748211

Vendo HP16C nuovissima, mai usata, ancora imballata sole L. 260.000 o cambio con HP15C. Ferdinando Auricchio - Via Orazio, 10 - 80123 Napoli - Tel. 081/660144

Cerco possessori TRS80 Mod. 2 e Mod. 3 per scambio informazioni ed esperienze. Luca Floris - Via G. Dezza, 20 - 00152 Roma - Tel. 06/5896049

Cerco joystick universale Cambridge Computing in qualsiasi condizione purchè funzionante. Roberto Morosi - Piazza IV Novembre, 9 - 20035 Lissone (MI) - Tel. 484901

Cerco "Personal Software" dal nr. 1 al nr. 16 o meglio le fotocopie degli articoli riguardanti il C64. Cambio con giochi per C64. Tratto possibilmente con utenti in zona Bologna. Fabrizio Nardi - Via Bonavia, 6 - 40068 S. Lazzaro di Savena (BO) - Tel. 46508

Occasione: vendo Xerox 820, doppio drive 8" più stampante margherita Xerox, CP/M, WordStar, emulazione 3270 a L. 6.000.000 più IVA. Software in omaggio. Telefonare ore ufficio. Ferruccio Ferrazza - Via Donati, 16 - 20146 Milano - Tel. 02/471528

Vendo serie programmi scientifici, matematici e finanziari per Sharp PC 1245 e 1251. Prezzi interessantissimi. Scrivere o telefonare ore pasti. Marco Pedroni - Via Mazzini, 22 - 26100 Cremona - Tel. 0372/37265

Vendo Enciclopedia di Scienza e Tecnica (E.S.T.) Mondadori più Enciclopedia di Elettronica e Informatica o cambio. Telefonare ore ufficio. Angelo Furia - Via A. Tealdi, 1/b - 56100 Pisa - Tel. 050/576613

Vendo per N.E. Z80 o Micro Design configurazione CP/M le schede aggiuntive "Computer Parlante" e "Grafica Sovrapposta" con relativo software. Eseguo schede personalizzate. Marco Sudetti - Via Martiri della Libertà, 52 - 10040 Borgaretto (TO) - Tel. 3580200

Vendo Sega SC-3000 32 Kbyte ancora in garanzia, perfetto, con circa 150 programmi a L. 35.000 non trattabili. Guglielmo Nervegna - Via Risorgimento, 273 - 47100 Forlì - Tel. 0543/84006

Compro hardware rotti, danneggiati, inutilizzabili, dei più conosciuti home computer. Annuncio sempre valido. Gian Luigi Calzolaro - Via M. della Benedicita, 3/12 - 16010 Rossiglione (GE) - Tel. 010/925447

Vendo Apple compatibile orientato alla grafica, completo di monitor, drive, video digitizer con telecamera, stampante e moltissimo software. Occasione d'oro. Tri - S.Polo 1674 - 30125 Venezia - Tel. 041/32382

## Guida per l'input dei programmi versioni VIC 20 e C 64

Notate che i listati contengono "parole" racchiuse tra parentesi graffe { }. Tali parole rappresentano caratteri di controllo come mostrato nel sottostante riquadro. Se sono precedute da un numero, questo indica il numero di volte che quel tasto deve essere premuto. Se il simbolo è sottolineato deve essere premuto contemporaneamente a SHIFT mentre se è racchiuso da [ < > ] deve essere premuto contemporaneamente al tasto COMMODORE. Inoltre, se tra parentesi si trova un carattere alfabetico "solitario", questo dovrà essere premuto contemporaneamente al tasto CONTROL.

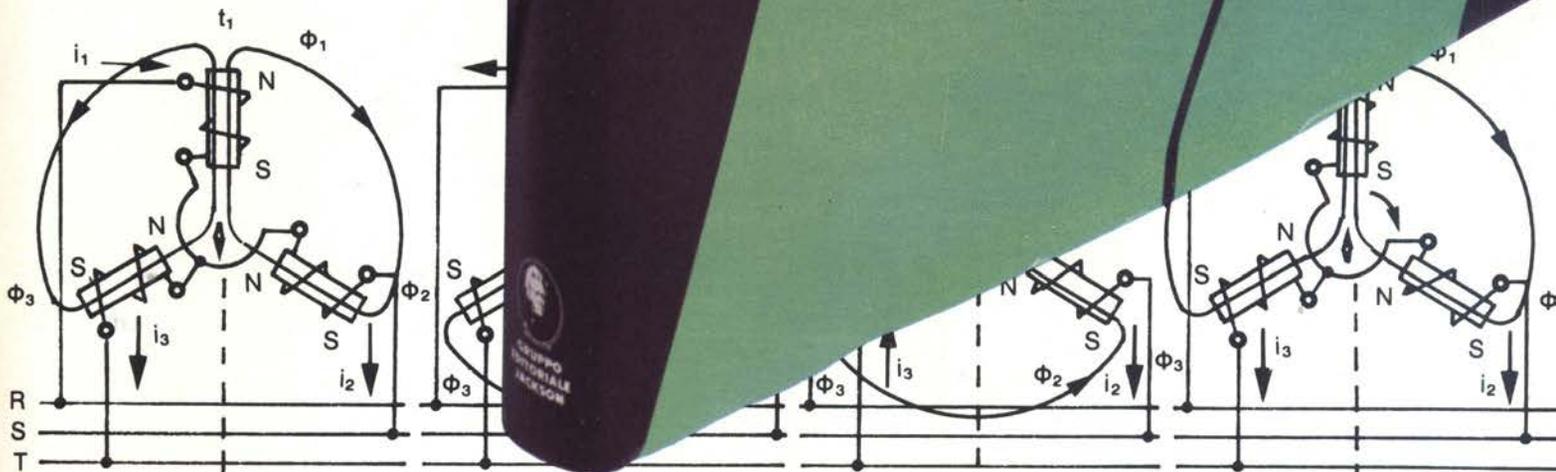
Con questo sistema di codifica, sarà molto più agevole copiare i listati senza faticose e dubbie interpretazioni di caratteri grafici e di controllo di cursore o dei colori.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	8	
{SU}	SHIFT    CRSR		{GRN}	CTRL 6		{F1}	F1	
{GIU'}	CRSR		{BLU}	CTRL 7		{F2}	F2	
{SIN}	SHIFT (=CRSR=>)		{YEL}	CTRL 8		{F3}	F3	
{DES}	(=CRSR=>)		[<1>]	7		{F4}	F4	
{RVS}	CTRL 9		[<2>]	7		{F5}	F5	
{OFF}	CTRL 0		[<3>]	7		{F6}	F6	
{BLK}	CTRL 1		[<4>]	7		{F7}	F7	
{WHT}	CTRL 2		[<5>]	7		{F8}	F8	
{RED}	CTRL 3		[<6>]	7				



Prenotazione riservata  
ai possessori della II edizione  
di E.I. Enciclopedia  
di Elettronica e Informatica.

PRONTO  
RILEGATO  
A PREZZO  
SPECIALE



Un volume di 100 pagine  
prezioso per lo studio, il  
lavoro, gli hobbies.  
Un'opera realizzata per  
chi vuol sapere tutto su:  
I circuiti in corrente  
continua,  
I circuiti in corrente  
alternata,  
Il magnetismo,  
l'Elettrostatica,  
I circuiti trifase.

# Elettrotecnica

*Per una visione completa del progresso,  
completa la tua E.I.*



**GRUPPO  
EDITORIALE  
JACKSON**

SAN FRANCISCO - LONDRA - MILANO

ENCICLOPEDIA  
DI ELETTRONICA  
& INFORMATICA

## **EI** RICHIESTA SPECIALE D'ACQUISTO

Si, per completare la II edizione  
di E.I. desidero ricevere il volume  
già rilegato di ELETTRONICA,  
al prezzo speciale di L. 29.000.

Da inviare in busta chiusa a:  
Gruppo Editoriale JACKSON  
Divisione Grandi Opere  
Via Rosellini, 12 - 20124 Milano



Allego:

- Assegno non trasferibile a voi intestato
- Fotocopia di vaglia a voi intestato
- Fotocopia di CCP a voi intestato

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Via \_\_\_\_\_

Città \_\_\_\_\_

Firma \_\_\_\_\_

CAP \_\_\_\_\_

Prov. \_\_\_\_\_

P. IVA (per le Aziende)

# Ritorna in edicola

# VIDEO BASIC

Il corso più entusiasmante su cassetta  
del Gruppo Editoriale Jackson per Commodore 64,  
VIC 20 e Spectrum

## 200.000 copie vendute

del 1° fascicolo della prima edizione

Ogni lezione  
uno spettacolo

Con la 1ª lezione  
una cassetta giochi  
compresa nel prezzo



Il corso è composto da:  
**20 fascicoli** + (Quattordicinali)  
**20 cassette** +  
**5 splendidi raccoglitori**

Oggi è davvero facile imparare il Basic. Con Video Basic il corso su cassetta che ti permette di programmare subito il tuo computer. È facile: tu chiedi, lui risponde, tu impari. Passo dopo passo. Sul tuo schermo appaiono le domande, le risposte, gli esercizi e

tu, senza fatica, presto e bene, impari a conoscere e programmare il tuo computer, sia esso un VIC 20, un Commodore 64 o un Sinclair. Video Basic è in edicola. Provalo subito. Ogni lezione è uno spettacolo.

Oggi il Basic si impara così. Video Basic, il corso su cassetta per parlare subito col tuo computer.

Video Basic  
per imparare non solo il Basic.



Un'altra grande idea firmata  
**GRUPPO EDITORIALE JACKSON**

Milano • San Francisco • Londra • Madrid