LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

ISSN 0392-8896

PERSONAL SOFTWARE

ANNO 4 N. 24
GENNAIO 1985 - L. 4.000

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

Spedizione in abb. postale Gruppo III/70

Copia riservata agli abbonati

COME PROGRAMMARE UN ADVENTURE CON IL C 64

TESTI PERSONALIZZATI CON LO SHARP

WORD PROCESSING PER TI99/4A

TORNEO DI BRIDGE CON APPLE

ALTA RISOLUZIONE CON C 64

TITOLAZIONE CON LO SPECTRUM

E' IN EDICOLA

Bit

Annuario hardware 1985

tutto l'hardware per l'informatica in Italia

Una realizzazione
GRUPPO EDITORIALE JACKSON
e
Istituto SISDOBDA

Home Computer
Micro e Personal Computer
Minisistemi - Supermini - Stampanti - Plotter

Supplemento a Bit Nr. 55/Novembre 1984 - Lire 8.000

Copertina fantastica per un programma fantastico: Dentro l'avventura, ovvero, come costruirsi un adventure game.

ARTICOLI

10 **TITOLAZIONE CON LO SPECTRUM** di *Paolo Coretti* — Spectrum

21 **WORD PROCESSOR PER TI99/4A** di *Stefano di Tommaso* — TI99/4A

35 **TRASFORMATE IL VOSTRO SPECTRUM IN UN REGISTRATORE DIGITALE** — Spectrum

36 **TESTI PERSONALIZZATI PER SHARP MZ 700** di *Giuseppe Gatti* — Sharp

46 **ALTA RISOLUZIONE** di *Antonio Bartalesi* — C 64

52 **GRAFICA E JOYSTICK PER APPLE II** di *Claudio Poma* — Apple

60 **TORNEO DI BRIDGE 2°** di *Vincenzo Delle Cave* — Apple

66 **ARCHIVIO PER SPECTRUM** di *Natale Carloni* — Spectrum

73 **COME AGGIUNGERE NUOVI COMANDI BASIC AL C 64** di *Maurizio Paolinelli* — C 64

82 **IMPARIAMO IL LINGUAGGIO MACCHINA CON IL VIC 20 E IL C 64 8°** di *Alessandro Guida* — VIC 20 - C 64

92 **DENTRO L'AVVENTURA 2°** di *Roberto Tabacco* — C 64

RUBRICHE

5 **EDITORIALE** di *Riccardo Paolillo*

6 **POSTA**

8 **PERSONAL NEWS** a cura di *Marco Giacobazzi*

I SEGRETI DEI PERSONAL:

96 I SUPER BASIC 2° di *Alessandro Guida* — VIC 20 - C 64

100 ALLA SCOPERTA DEL LINGUAGGIO MACCHINA di *Mauro Lenzi* — Sharp

102 L'USO E... L'ABUSO DEI CARATTERI DI CONTROLLO di *Marcello Spero* — Spectrum

105 RING di *Sergio Borsani* — TI99/4A

112 **PICCOLI ANNUNCI**

GUIDA

N. 24
GENNAIO 1985

è in edicola il nuovo numero

SPECIALE SOFTWARE C 64

- BITEST: ALPHA-MICRO
- SOFTEST: ACCESS II
- IN PROVA: C 64 EXECUTIVE PLUS/4
- ORECCHIO MUSICALE CON IL SEGA
- L'ELETTROSTATICA E L'APPLE
- LO SPECTRUM PARLANTE
- CIAPA NO CON C 64

CON INSERTO:

SUPER BIT RISERVATO PERSONAL

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

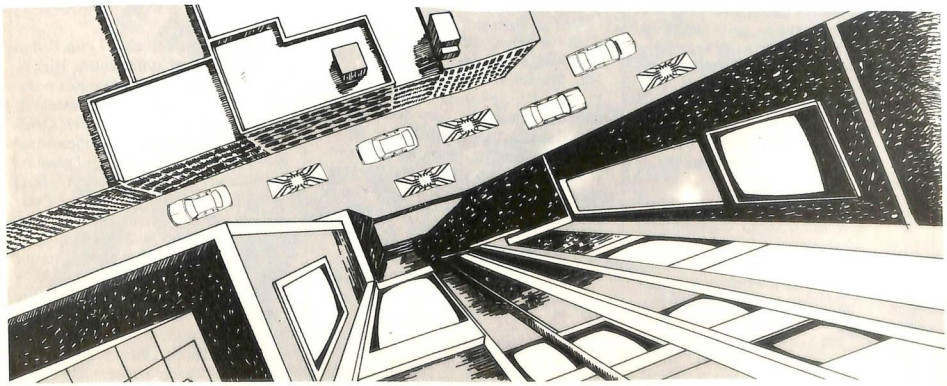
EDITORIALE

La città informatica

di Riccardo Paolillo

Si è spesso osservato come il proliferare di nuovi computer shop stia rapidamente cambiando il panorama commerciale delle nostre città.
Questo processo è ulteriormente accentuato da nuove iniziative che modificano lentamente, ma sensibilmente, il nostro modo di vivere. Pur senza addentrarci in discorsi sociologici, che sarebbero sicuramente fuori luogo, ci sembra opportuno esaminare questi fenomeni che coinvolgono, anche se marginalmente, la vita di ognuno di noi.
Il computer shop rimane, ovviamente, il polo maggiore di attrazione: con le sue vetrine, le sue luci e i calcolatori in funzione esercita il fascino irresistibile della tecnologia alla portata di tutti, ma basta spostarsi di pochi passi e scoprire altri elaboratori in luoghi imprevedibili.
Tutte le ricevitorie del Totocalcio, ad esempio, sono dotate di apparecchi per la convalida delle giocate, che altro non sono che piccoli calcolatori opportunamente programmati. Spesso poi sono affiancati da personal veri e propri, con i quali i gestori possono rapidamente effettuare sviluppi di sistemi ridotti da proporre ai clienti.
Un altro esempio di informatica a disposizione di tutti è quello degli sportelli bancari automatici ormai diffusissimi dappertutto. Chi li usa, e ne apprezza l'utilità, forse non si rende neanche conto che in realtà si tratta di calcolatori elettronici.
Questi semplici esempi, riscontrabili nella nostra vita quotidiana, dimostrano che l'incontro uomo computer è già una realtà, e non si tratta di un evento necessariamente traumatico come alcuni vogliono far credere.
Molto più facilmente si deve prendere atto che il calcolatore, specie se personale, può essere utilizzato non solo per applicazioni gestionali o come strumento di videogame. A Milano è nato recentemente un centro per l'interscambio personale di annunci economici; gli strumenti di trasferimento delle informazioni tra i vari inserzionisti, sono proprio dei personal computer che oltre a consentire accessi rapidi a molti dati, permettono di avvicinare gli utenti (soprattutto giovani) al mondo affascinante dell'informatica popolare.
Tutte queste iniziative contribuiscono a rallegrare ed animare le strade dei centri cittadini: sicuramente, in questi casi, il personal agisce da catalizzatore e aiuta a rafforzare e migliorare i rapporti interpersonali.

Bisogna solo augurarsi che questa tendenza non porti a situazioni esagerate e si pretenda di automatizzare attività non idonee: il rifiuto che sicuramente ne scaturirebbe, cancellerebbe rapidamente questi tentativi di presentare in modo amichevole e utile le nuovissime tecnologie.



Un menu personalizzato

Sono un utente di Commodore 64, che si rifiuta di usare la macchina solo per scopi futili, bensì prova a sviluppare delle possibili applicazioni, che cerchino di migliorare le possibilità della stessa.
A questo scopo vorrei formulare un paio di domande, per le quali aspetto vostra gentile risposta.
È noto che, all'accensione, il computer va alla ricerca di una locazione di memoria, ne legge il contenuto, e si comporta di conseguenza.
Gentilmente vi chiedo:
1) la locazione di memoria iniziale;
2) quale linguaggio usare, per modificare il contenuto di questa locazione;
3) nel caso risultasse il linguaggio macchina, data la mia totale inesperienza; vi pregherei di scrivermi una piccola routine in tale linguaggio, che chieda una sigla di riconoscimento, facendone apparire la domanda e verificandone l'esattezza della risposta. Se quest'ultima risultasse affermativa fare in modo che passi allo schermo normale, mentre in caso negativo continui a richiederla. Vi prego inoltre di immettere nella stessa routine, come condizione iniziale, che lo schermo e il bordo diventino neri, mentre il cursore verde.

Fabio Fracas
Rimini (FO)

Al momento dell'accensione del C 64, il sistema esegue automaticamente alcune operazioni e, in particolare, controlla la presenza di una cartuccia per l'avviamento automatico della ROM, nella locazione decimale 32768. Se la cartuccia è presente, e i primi nove byte sono codificati secondo un ben preciso standard, l'inizializzazione normale viene sospesa e il controllo viene trasferito al codice della cartuccia; in caso contrario, continua il normale sistema di inizializzazione. In questo modo si spiega perché non occorra dare particolari comandi di RUN o SYS, quando si accende il calcolatore con una cartuccia inserita.

Questo sistema di tipo hardware rende molto problematica la realizzazione di quanto da lei desiderato: occorrerebbe infatti realizzare una cartuccia per eseguire un programma di inizializzazione opportuno.

Una soluzione molto più semplice, anche se incompleta, in quanto non eseguibile automaticamente all'accensione, è quella di realizzare un piccolo programma BASIC da mandare in esecuzione subito dopo l'avvio mediante i tasti SHIFT e RUN: questo programma potrebbe poi essere registrato all'inizio di ogni nastro.

Ottocentisti cercasi

Vorrei chiedervi se sapete dove e cosa reperire di specifico per istruirmi all'uso totale del mio 800 XL (possibilmente in italiano). Questo è un problema sentito anche da numerosi conoscenti che con l'ausilio della guida per l'800 (unica reperibile in italiano e solo parziale) non potranno mai "conoscere" a fondo l'800 XL. Potete indirizzarci? Grazie se pubblicate la presente. Invito i più esperti a farci sapere o inviare qualcosa.

Nello Scalia
Torino

Non possiamo fare altro che rilanciare il suo appello. L'Atari 800 XL è una macchina relativamente "giovane", ma ci risulta abbia già incontrato il favore di molti appassionati. Coraggio quindi: aspettiamo le vostre collaborazioni, per cominciare a parlare anche del nuovo Atari su **Personal Software**.

Programmi d'autore

Sono un vostro nuovo lettore e nel numero di Agosto-Settembre, nel testo dell'articolo *Rally* di Barzaghi, ho letto dei riferimenti ai programmi *La 24 ore di Le Mans, Formula 1, Gran Prix, Il campionato di calcio di Serie A e B, All that basket*. Gradirei mi comunicaste per ciascuno dei programmi citati la rivista ed il rispettivo numero su cui sono stati pubblicati i listati (ed anche quello delle eventuali correzioni ai listati). Vorrei inoltre sapere se sono compatibili con il Commodore 64 e se sono salvabili su nastro.

Leo Loria
Torino

Tutti i programmi da lei citati sono stati pubblicati sulla rivista Bit. *Precisamente e in ordine di uscita, abbiamo:* Il campionato di calcio di Serie A e B *sul numero 11 dell'Ottobre 1980.* Le Mans, *Pubblicato sul mero 26 del Marzo '82 e sul numero* 1 *sul numero 18 del Giugno '81. Infine* All that basket e Grand Prix, *pubblicati rispettivamente sul numero 26 del marzo '82 e sul numero 27 dell'Aprile '82. Tutti i programmi sono stati scritti per PET/CBM. e non sono quindi direttamente utilizzabili sul C 64. Riteniamo comunque che, data la somiglianza dei linguaggi delle due macchine, e l'abbondanza di spiegazioni con cui Bar-*

zaghi accompagna i propri articoli, non sia eccessivamente complicato ricavarne delle versioni per il Commodore 64.

■ ■ ■

I caratteri scomparsi

Posseggo un C 64 da alcuni mesi, ma solo di recente ho potuto impegnarmi a fondo nell'apprendimento del linguaggio BASIC e nella programmazione. Tutt'ora sono agli inizi, ma sono sicuro che più uno prova a programmare, tanto più imparerà.
Da un po' di tempo sto studiando un programma con le funzioni di macchina per scrivere (si impostano i dati ed il testo e la stampante lo esegue), ma purtroppo, sono rimasto incagliato poiché la stampante si rifiuta di stamparmi le virgole nel testo. Scrive fino alla virgola, ma ignora insieme ad essa, tutto il testo che segue, spezzettando le frasi e rendendo il testo ridicolo.
Ho provato a risolvere il problema in diversi modi, ma un po' per ignoranza e magari sfortuna, non l'ho mai imbroccata. Per cui chiedo se potreste aiutarmi voi e consigliarmi cosa devo fare per risolvere questo problema.
Leggo **Personal Software** da poco, ma lo trovo molto interessante ed utile e devo ammettere che i vostri listati per il momento mi fanno molto comodo.

Riccardo Di Ciammo
Milano

Anche se non ci ha inviato il suo programma, riteniamo di aver individuato l'origine del suo problema. Molto probabilmente non si tratta di un malfunzionamento della stampante, ma di un uso non completamente corretto dell'istruzione INPUT del linguaggio BASIC. Questa istruzione, utilizzata per assegnare dei valori alle variabili del programma, non consente di utilizzare alcuni caratteri e in particolare la virgola.

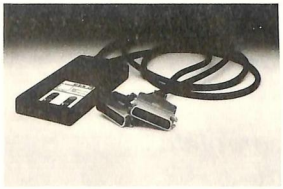
Il motivo di questa limitazione risiede nel fatto che la sintassi del comando prevede la possibilità di effettuare assegnamenti a più variabili, separate da virgole, con un'unica istruzione di INPUT. Una eventuale virgola introdotta da tastiera viene quindi interpretata come separatore tra una variabile e la successiva, che in realtà non esiste. Per aggirare questo ostacolo, visto che in un programma come il suo è necessario poter introdurre anche le virgole, occorre utilizzare l'istruzione GET anziché la INPUT. L'istruzione GET accetta un unico carattere, compresa la virgola. A differenza della INPUT i caratteri introdotti non vengono visualizzati sullo schermo e non occorre premere il tasto RETURN. È comunque molto facile scrivere una semplice routine che, visualizzando mediante una PRINT tutti i caratteri via via introdotti, simuli perfettamente l'istruzione INPUT, mediante l'uso di una serie di GET successive.

E'IN EDICOLA
PC MAGAZINE
La rivista dei sistemi MS-DOS
La guida completa del personal computer IBM e compatibili
con tutta la competenza del

GRUPPO EDITORIALE JACKSON

Metti un turbo alla stampante...

Anche per il mondo dei micro è arrivato il momento di "liberare" la CPU dai compiti riguardanti il completamento delle operazioni di input/output con la periferia. Nel campo dei mainframe ciò è avvenuto attraverso graduali passaggi, dal "canale" che sincronizzava il flusso delle informazioni in entrata ed in uscita, ad un'architettura "multiprocessor" con CPU specializzate per i vari compiti, pronte a dialogare con l'intelligenza propria delle periferiche. Il primo passo per i personal è compiuto dai costruttori di hardware che possono dotare i propri dispositivi di un buffer interno che consente un certo "parallelismo" delle operazioni. Nel caso delle stampanti questa memoria "locale"

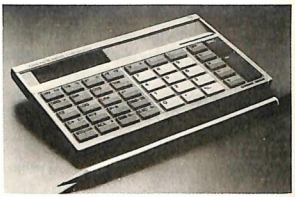


è spesso però incapace di contenere un intero tabulato di una certa dimensione, vanificando così il presunto guadagno legato alla presenza del buffer. Esistono perciò dispositivi ausiliari che agiscono su due fattori critici: come nel caso dell'interfaccia CTX, contengono un banco di memoria di transito ampliabile in loco (a partire da 8 Kbyte per giungere a 256) e possono ricevere dati ad una velocità massima di 38.400 baud, realmente proibitiva per molte delle stampanti in commercio. L'altro lato dell'interfaccia opera la conversione alla velocità del dispositivo di output. La CTX si presenta inoltre in tre versioni, per consentire l'uso di stampanti con uscita seriale anche ai micro dotati di interfaccia parallela o viceversa.

Micro Nord
Viale Principe Eugenio, 32
39100 Bolzano
Tel. 0471-30144

La Texas propone Galaxy

L'ultima calcolatrice scientifica della Texas Instruments si chiama TI-30 Galaxy ed è stata progettata privilegiando le applicazioni didattiche, vale a dire la completezza delle funzioni, la robustezza e la praticità d'uso. Obiettivo della Texas è di rendere la Galaxy un nuovo punto di riferimento per gli studenti delle scuole secondarie. La tastiera comprende 42 tasti di ampie dimensioni a rilievo, suddivisi in quattro aree funzionali: numerica, operazioni, esecuzione e accensione. Il display è inclinato e dà indicazione delle operazioni in corso o delle funzioni attivate. Dotata del sistema operativo AOS (Algebric Operating System) la Galaxy consente 15 livelli di parentesi e può mantenere sospese fino a quattro operazioni. Provvista di memoria costante, la nuova Texas dispone delle operazioni più comuni (funzioni trigonometriche, esponenziali e logaritmiche, radici e potenze) e di funzioni statistiche e di conversione da gradi a radianti nonchè dal sistema gradi/minuti/secondi al sistema decimale, oltre alle conversioni polari.



Texas Instruments
Viale Europa, 38/44
20093 Cologno Monzese (MI)
Tel. 02-2532451

E' arrivato anche il QL

Dovrebbero essere ormai cominciate le consegne del Quantum Leap della Sinclair, l'ultima rivoluzionaria creatura di Sir Clive che tanto si è fatta attendere dagli appassionati che avevano inoltrato i primi ordini postali in Inghilterra. La commercializzazione viene curata dalla Re-

bit Computer, una divisione della GBC di Cinisello Balsamo, distributore esclusivo dei prodotti Sinclair in Italia. Dai primi giorni di quest'anno è a disposizione anche la versione italiana della tastiera e dei programmi integrati, che sono stati tradotti per il bene di chi non conosce l'inglese. La Rebit ha allo studio una forma di garanzia per l'utente, simile a quella già sperimentata per lo Spectrum e ha potenziato il proprio servizio di assistenza tecnica per coprire le esigenze dei "Sinclairisti" con tempestività, su tutto il territorio nazionale.

Columbia,
Largo Toscanini, 1
20100 Milano
Tel. 02-7798267

Non tutto è perduto

Una interessante novità per hobbysti e utenti professionali alle prese con i disk drive Commodore formato D4 (compreso il Petdrive da 170 Kbyte) e 1541 ci giunge dall'Inghilterra. La Mermaid Software annuncia uno strumento capace di recuperare tutto il possibile da dischi danneggiati e illeggibili con i normali mezzi a disposizione. Doctadisk è un programma che consente di ritrovare dati alterati o presenti su dischi guastati dall'età o da eventi dannosi, trasferendoli su un disco nuovo. Come se non bastasse è possibile copiare la testata del disco difettoso, vedere il contenuto di tracce e settori mentre si leggono i dati e riutilizzare il disco originale, se non era stato fisicamente alterato. Doctadisk permette anche di ottenere veloci copie di backup, il tutto per meno di 16 sterline (compreso un disco vergine). La Mermaid Software ricerca distributori per l'Italia.

Mermaid Software
27 Buckingham Gardens,
Edgware, Middlesex
HA8 6NB
England
Tel. (01) 951 3355
Telex 923574 ALACOL G

Pagine del software per Apple

La Editronica ha realizzato un catalogo comprendente un migliaio di programmi per i personal computer Apple. In 274 pagine sono illustrate le applicazioni reperibili sul mercato italiano per le macchine prodotte dalla casa di Cupertino, dal fortunato modello II al Macintosh, non tralasciando il recente IIc. Le schede di corredo riportano le notizie necessarie per una precisa identificazione: mercato dell'applicazione, prezzo, produttore, configurazione hardware richiesta. E' stato altresì ideato un "servizio lettori" per ottenere a stretto giro di posta informazioni più dettagliate o particolari direttamente dalle case produttrici di software.

Editronica
Corso Monforte, 39
20122 Milano
Tel. 02-702803

Software Arts e VisiCorp firmano l'armistizio

Ponendo fine ad una serie di poco onorevoli litigi, le due società americane hanno raggiunto un pacifico accordo riguardante la proprietà, lo sviluppo e la vendita del VisiCalc, il primo foglio elettronico per personal computer largamente diffuso in tutto il mondo. Il software fu realizzato nel 1979 da Software Arts, che cedette in quello stesso anno i diritti di vendita alla VisiCorp che gli impose il proprio marchio VisiCalc. Secondo quanto stabilito nel protocollo d'intesa, VisiCorp non distribuirà più il package. Il foglio elettronico torna così sotto le ali della società creatrice che ha inoltre ottenuto dalla VisiCorp la lista dei clienti ed il diritto a riprodurre i manuali. Il recente spreadsheet della VisiCorp si chiamerà d'ora in avanti Visi On Plan e non più Visi On Calc, ma non sono lesi i diritti di distribuzione di FlashCalc, prodotto da una terza software house. Dulcis in fundo, VisiCorp rifonderà mezzo milione di dollari a Software Arts per le precedenti vendite di VisiCalc.

VisiCorp
Software Arts

PERSONAL SOFTWARE

Titolazione con lo Spectrum

Un aiuto nell'aula di chimica

di *Paolo Coretti*

Il lavoro qui presentato permette, tramite lo studio del pH (grado di acidità o basicità delle soluzioni), di simulare una titolazione.
La struttura del listato è lineare e non presenta difficoltà di comprensione: questo è un tipico programma in cui la parte più laboriosa è la risoluzione degli algoritmi di calcolo, piuttosto noiosa normalmente, ma rapida con il computer.
Dopo la visualizzazione sul display del menu principale (linee 2-18), si opera la scelta delle opzioni disponibili (righe 19-34).
Le opzioni sono:
1 - calcolo del pH di una soluzione di acido forte (righe 100-102);
2 - calcolo del pH di una soluzione di base forte (righe 105-111);
3 - calcolo del pH di una soluzione di acido debole (riga 120);
4 - calcolo del pH di una soluzione di base debole (riga 130);
5 - calcolo del pH di una soluzione di idrolisi (righe 150-220);
6 - simulazione della titolazione di una soluzione di acido debole con una base forte (righe 250-390);
7 - calcolo del pH di una soluzione formata da un certo volume di un acido debole a concentrazione nota con una base forte di volume e con-

Listato 1. *Il programma per lo Spectrum.*

```
  2 CLS : PRINT AT 0,7;"Analisi
 del pH"
 10 PRINT AT 7,0;"1-Acido forte
"
 11 PRINT "2-Base forte"
 12 PRINT "3-Acido debole"
 13 PRINT "4-Base debole"
 14 PRINT "5-Idrolisi"
 15 PRINT "6-Titolazione"
 16 PRINT "7-Analisi titrimetri
ca"
 17 PRINT "8-Titolazione acido
diprotico"
 18 PRINT : PRINT : PRINT ; PRI
NT "Battere il tipo di scelta":
PAUSE 0
 19 IF INKEY$="f" THEN STOP
 20 IF INKEY$="1" THEN GO TO 10
0
 22 IF INKEY$="2" THEN GO TO 11
0
 24 IF INKEY$="3" THEN GO TO 12
0
 26 IF INKEY$="4" THEN GO TO 13
0
 28 IF INKEY$="5" THEN GO TO 15
0
 30 IF INKEY$="6" THEN GO TO 25
0
 32 IF INKEY$="7" THEN GO TO 40
0
 34 IF INKEY$="8" THEN GO TO 50
0
 100 INPUT "C=";c: IF c>1e-6 THE
N GO TO 102
 101 LET h=-LN ((c+SQR (c*c+4*1e
-14))/2): LET h=h/LN 10: PRINT "
pH=";h: STOP
 102 LET h=-LN c: LET h=h/LN 10:
 PRINT "pH=";h: STOP
 105 LET h=-LN c/LN 10: LET h=14
-h: PRINT "pH=";h: STOP
 110 INPUT "C=";c: IF c>1e-6 THE
N GO TO 105
 111 LET h=-LN ((c+SQR (c*c+4*1e
-14))/2)/LN 10: LET h=14-h: PRIN
T "PH=";h: STOP
```

```
 120 PRINT : INPUT "C=";c: PRINT
 "C=";c: INPUT "Ka=";k: PRINT "K
a=";k: LET h=-LN (SQR (k*c))/LN
10: PRINT "pH=";h: STOP
 130 PRINT : INPUT "C=";c: PRINT
 "C=";c: INPUT "Kb=";k: PRINT "K
b=";k: LET h=-LN (SQR (k*c))/LN
10: PRINT "pH=";14-h: STOP
 150 PRINT : PRINT "Ka o Kb<a/b)
?": PAUSE 0: IF INKEY$="b" THEN
GO TO 220
 200 PRINT : INPUT "Ka=";k: PRIN
T "Ka=";k: INPUT "C=";c: PRINT "
C=";c: LET h=-LN (SQR ((1e-14/k)
*c))/LN 10: PRINT "pH=";14-h: ST
OP
 220 PRINT : INPUT "Kb=";k: PRIN
T "Kb=";k: INPUT "C=";c: PRINT "
C=";c: LET h=-LN (SQR ((1e-14/k)
*c))/LN 10: PRINT "pH=";h: STOP
 250 CLS : PRINT AT 0,0;"Titolaz
ione acido debole-base": PRINT "
          forte"
 260 PRINT : PRINT : INPUT "C.ac
ido=";c: PRINT "C.acido=";c: INP
UT "Vol.acido=";h: PRINT "Vol.ac
ido=";h: INPUT "Ka=";k: PRINT "K
a=";k: INPUT "C.base=";e: PRINT
"C.base=";e: INPUT "Var.volume="
;d: PRINT "Var.volume=";d
 270 PRINT : PRINT : PRINT "Anal
isi o grafico(a/g)?": PAUSE 0: I
F INKEY$="g" THEN GO SUB 900: GO
 TO 900
 275 CLS
 280 LET z=-LN (SQR (k*c))/LN 10
: LET z=(INT (z*100)/100): PRINT
 "pH iniziale=";z
 290 LET a=0
 300 LET a=a+d
 305 LET b=k*c-a*e
 310 IF b<=0 THEN GO TO 330
 315 LET p=-LN (k*b/(a*e))/LN 10
 316 LET p=INT (100*p)/100
 320 PRINT "Volume=";a;"    ";"pH
 tampone=";p
 325 GO TO 300
 330 LET q=(a*e)/(h+a)
 340 LET p=14+LN (SQR ((1e-14)/k
*q))/LN 10
 345 LET p=INT (100*p)/100
```

Titolazione con lo Spectrum

```
 350 PRINT "Volume=";a;"  ";"pH
idrolisi=";p
 355 LET i=a+h: LET a=0
 360 LET a=a+d
 365 LET p=14+LN (((a*e)/(i+a)))
/LN 10
 366 LET p=INT (p*100)/100
 370 PRINT "Vol.base eccesso:";a
;TAB 22;"pH=";p
 380 IF a<=h-1 THEN GO TO 360
 390 LET g=(INT (p+1)): PRINT "p
H asintotico=";g: PAUSE 0: IF IN
KEY$="f" THEN STOP
 391 GO SUB 300: GO TO 950
 400 CLS : INPUT "C.acido=";c: P
RINT "C.acido=";c: INPUT "Ka=";k
: PRINT "Ka=";k: INPUT "V.acido=
";v: PRINT "V.acido=";v: INPUT "
C.base=";e: PRINT "C.base=";e: I
NPUT "V.base=";d: PRINT "v.base=
";d
 402 LET b=(v*c)-(d*e): LET i=(d
*e)/(v+d)

 403 IF b>0 THEN GO TO 406
 404 IF b=0 THEN GO TO 407
 405 IF b<0 THEN GO TO 410
 406 LET h=-LN ((k*b)/(d*e))/LN
10: LET h=INT (h*100)/100: PRINT
 : PRINT : PRINT "Tampon
e a pH=";h: PAUSE 0: GO TO 1
 407 LET w=SQR ((1e-14/k)*i): LE
T h=14+LN w/LN 10: LET h=INT (h*
100)/100: PRINT : PRINT : PRINT
 "Idrolisi:pH=";h: PAUSE
0: GO TO 2
 410 LET w=(ABS b)/(v+d): LET h=
14+LN w/LN 10: LET h=INT (h*100)
/100: PRINT : PRINT : PRINT
INT "pH=";h: PAUSE 0: GO TO 2
 411 STOP
 500 CLS : PRINT AT 0,0;"Esempio
 di titolazione di acido": PRINT
 "          diprotico"
 501: PRINT "          (100 ml)

 512 PRINT : PRINT : PRINT : PRI
NT : INPUT "K1=";k: PRINT "K1=";
k: INPUT "K2=";w: PRINT "K2=";w:
 INPUT "C.acido=";c: PRINT "C.ac
ido=";c: INPUT "V.base=";v: PRIN
T "V.base=";v
 640 LET m=0
 642 PRINT : PRINT : PRINT "graf
ico(s/n)?": PAUSE 0: IF INKEY$="
s" THEN GO TO 750
 650 CLS : LET h=SQR (k*c): LET
h=-LN h/LN 10: LET h=INT (h*100)
/100: PRINT "pH iniziale=";h
 660 LET m=m+v: PRINT "Vol.base=
";m: IF m=50 THEN GO TO 890
 670 LET g=(100-m*2)/(m*2): LET
h=k*g: LET h=-LN h/LN 10: LET h=
INT (h*100)/100: PRINT "pH =";h
 680 IF m<50 THEN GO TO 660
 690 LET h=SQR (k*w): LET h=-LN
h/LN 10: LET h=INT (h*100)/100:
PRINT "I punto di equivalenza:pH
=";h
 700 LET m=m+v: PRINT "V.base=";
m: IF m=100 THEN GO TO 730
 710 LET g=(100-m)/(m-50): LET h
=w*g: LET h=-LN h/LN 10: LET h=I
NT (100*h)/100: PRINT "pH=";h
 720 IF m<100 THEN GO TO 700
 730 LET h=SQR ((1e-14*c)/w): LE
T h=14+LN h/LN 10: LET h=INT (10
0*h)/100: PRINT "II punto di equ
ivalenza:pH=";h: PAUSE 0: IF INK
EY$="f" THEN STOP
 740 GO TO 2
 750 GO SUB 900
 799 LET m=0
 800 LET h=SQR (k*c): LET h=-LN
h/LN 10: LET x=m: LET h=h*10: PL
OT x+50,h+30
 810 LET m=m+v: LET x=m: IF m=50
 THEN GO TO 840
 820 LET g=(100-m*2)/(m*2): LET
h=k*g: LET h=-LN h/LN 10: LET h=
h*10: PLOT x+50,h+30
 830 IF m<50 THEN GO TO 810: LET
x=50
 840 LET h=SQR (k*w): LET h=-LN
h/LN 10: LET h=h*10: PLOT x+50,h
+30
 850 LET m=m+v: LET x=m: IF m=10
0 THEN GO TO 880
 860 LET g=(100-m)/(m-50): LET h
=w*g: LET h=-LN h/LN 10: LET h=h
*10: PLOT x+50,h+30
 870 IF m<100 THEN GO TO 850
 880 LET h=SQR ((1e-14*c)/w): LE
T h=14+LN h/LN 10: LET x=100: LE
T h=h*10: PLOT x+50,h+30
 885 PAUSE 0: IF INKEY$="f" THEN
 STOP
 886 GO TO 2
 899 STOP
 900 CLS : PLOT 50,30: DRAW 200,
0: PLOT 50,30: DRAW 0,140
 901 FOR x=50 TO 250 STEP 10: PL
OT x,30: DRAW 0,2: NEXT x
 902 FOR y=30 TO 170 STEP 10: PL
OT 50,y: DRAW 2,0: NEXT y
 903 PRINT AT 19,6;"0": PRINT AT
 19,12;"50": PRINT AT 19,18;"100
": PRINT AT 19,29;"200"
 904 PRINT AT 9,4;"7": PRINT AT
0,4;"pH": PRINT AT 21,17;"ml": P
RINT AT 9,0;"pH"
 905 RETURN
 950 LET z=-LN (SQR (k*c))/LN 10
: LET x=0: LET z=z*10: PLOT x+50
,z+30
 960 LET a=0
 970 LET a=a+d: LET x=a
 980 LET b=h*c-a*e
 990 IF b<=0 THEN GO TO 1020
1000 LET p=-LN (k*b/(a*e))/LN 10
: LET p=p*10: PLOT x+50,p+30
1010 GO TO 970
1020 LET b=c*e/(c+e)
1030 LET p=14+LN (SQR (1e-14/k*b
))/LN 10: LET p=p*10: PLOT x+50,
p+30
1040 LET i=a+h: LET a=0
1050 LET a=a+d: LET x=a+h
1060 LET p=14+LN ((a*e)/(i+a))
/LN 10: LET p=p*10: PLOT x+50,p+
30
1080 IF a<=h-1 THEN GO TO 1050
1090 LET g=(INT p+1): LET y=g: P
LOT (2*h+50),g+30: PAUSE 0: IF I
NKEY$="f" THEN STOP
1091 CLS : GO TO 250
```

E' IN EDICOLA

Videogiochi

CON TUTTA LA COMPETENZA DEL

GRUPPO EDITORIALE JACKSON

LA PRIMA
E UNICA RIVISTA DI
VIDEOGAMES
GIOCOMPUTER
GIOCHI ELETTRONICI

1) $C_a = \frac{V_1 N_1}{V_1 + V_2}$ $C_b = \frac{V_2 N_2}{V_1 + V_2}$

C_a = Concentrazione acido
C_b = Concentrazione base
V_1 = Volume dell'acido all'inizio della titolazione.
V_2 = Volume della base all'inizio della titolazione.
N_1 = Concentrazione acido.
N_2 = Concentrazione base.

2) Soluzione tampone:

$$C_a = \frac{V_1 N_1 - V_2 N_2}{V_1 + V_2} \qquad C_b = \frac{V_2 N_2}{V_1 + V_2}$$

$$[H^+] = K_a \frac{C_a}{C_b} \qquad [H^+] = K_a \frac{V_1 N_1 - V_2 N_2}{V_2 N_2}$$

V_1 = Volume iniziale acido - volume base aggiunta.
V_2 = Volume base dopo le varie aggiunte.
$B = V_1 N_1 - V_2 N_2$. Se $B = 0$, si avrà l'idrosili, e la concentrazione sarà:

3) $\frac{V_2 N_2}{V_1 + V_2}$ in cui V_2 è il volume della base aggiunta nel momento dell'idrolisi.

Superato il punto di idrolisi, avremo:

4) $[OH^-] = \frac{V N}{(V_a + V_b) + V}$

V = Volume della base aggiunta.
N = Concentrazione base.
V_a = Volume dell'acido all'idrolisi.
V_b = Volume della base all'idrolisi.

Infine, per il calcolo dell'opzione 7, avremo, nel caso di base in eccesso rispetto all'acido:

5) $[OH^-] = \frac{|V_1 N_1 - V_2 N_2|}{V_1 + V_2}$

Figura 1. *L'insieme degli algoritmi di calcolo.*

Leoni informatica

HARDWARE & SOFTWARE HOUSE

HOTLINE

linea telefonica dedicata alla risoluzione dei problemi dei clienti. Chiamando il numero telefonico riservato che troverete sulla cartolina garanzia acclusa ai programmi, riceverete tutte le informazioni che vi necessitano.

UPDATE

servizio di aggiornamento continuo dei programmi acquistati. Ogni modifica ai programmi realizzati dalla Leoni Informatica sarà fornita agli utenti degli stessi.

GARANZIA

tutti i programmi Leoni Informatica sono coperti da garanzia a Vita contro guasti di origine.

COMMODORE 64 SOFTWARE

SOFTWARE PER COMMODORE 64 E PLUS/4

Il presente listino entra in vigore dal 25 settembre 1984

Programmi in configurazione base (*) IVA esclusa

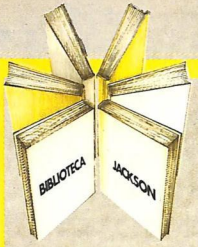
Cod.	Descrizione	Prezzo	Cod.	Descrizione	Prezzo
		PERSONALI			
0046	Ammortamento Mutui	60.000	0049/T	Totoplus	100.000
0050/T	Totocalcio a sviluppo colonnare	80.000	0051/T	Gestione dei Conti di casa	100.000
0058/T	Calcolo dell'equo canone	80.000	0055/T	Impariamo il BASIC	100.000
0059/T	Modello 740	10.0000	0063	Cento programmi BASIC	80.000
0066	Conto corrente	100.000	0091/T	Rubrica-telefonica	100.000
0176	Diary 64 (Commodore)	95.000	0174/T	Corso di Dattilografia	80.000
		GESTIONI GENERALI			
0047	Anagrafiche	150.000	0056	Dichiarazione I.V.A.	60.000
0065/T	Fido Clienti	100.000	0067	Piano dei Conti (cli/for/gen)	100.000
0068	Appuntamenti	100.000	0071	Ordini (cli/for)	100.000
0090	Mailing List (riordino alfabetico cap. prov.)	100.000	0094	Scheda 4800 car. (cli/for/rapp./az. etc.)	100.000
0096	Scheda 4800 car. agganciata al Mailing	150.000	0097	Super Mail (5 chiavi accesso, riordini..)	180.000
0116	Scadenziario effetti (ric. bancarie, tratte, etc.)	100.000	0120	Contabilità fatture (Iva, impon. etc.)	100.000
0121	Contabilità Semplice (Tratte/Fatt./Conti/etc.)	250.000	0160	Bolle e Fatture	200.000
0125	Contabilità Generale (132 colonne)	300.000			
		GESTIONI SPECIFICHE			
0045	Agenti e Rappresentanti	150.000	0048	Scadenziario premi e polizze	150.000
0164	Agenzie Immobiliari	150.000	0086	Librerie e biblioteche	120.000
0148	Studi Ottici	200.000	0151	Farmacie	300.000
0149	Studi Dentistici	200.000	0152	Studi Medici	200.000
0131	Hotel e Pensioni	280.000	0132	Parrucchieri	280.000
0133	Gommisti	280.000	0134	Clubs Nautici	280.000
0135	Officine	280.000	0171	Ristoranti	280.000
0170	Tavola Calda	280.000	0172	Lavanderie	280.000
0175	Condominio New (132 colonne)	400.000			
		GESTIONE TESTI			
0190	Hes Writer	70.000	0191	Word Processor III	100.000
0192	Bank Street Writer	70.000	0319	Easy Script/T (Commodore)	75.000
		TECNICI			
0136	Legge 373 (calcolo degli isolamenti termici)	100.000	0140	Ingegneria civile I (calcoli strutt.)	100.000
0141	Ingegneria civile II (travi intelaiate)	100.000	0404	Computo metrico	200.000
0409	Diagnostica C64	40.000	0322	Doctor 64	60.000
		MAGAZZINI			
0142	Magazzino e Fatturazione semplici	100.000	0143	Magazzino Grossisti (2500 art.)	280.000
0144	Magazzino e Fatturazione agganciati	200.000	0158	Magazzino Dettaglio (2500 art.)	280.000
0148	Magazzino codice alfanumerico (600 art.)	200.000	0159	Magazzino Taglia/Col. (2500 art.)	280.000
		LINGUAGGI & UTILITIES			
0162/T	Screen Grafix (Abacus)	85.000	0163	Copia Disco singolo	50.000
0064	Petspeed Compiler (Commodore)	80.000	0165/T	Zoom	70.000
0167	Simon's Basic (Commodore)	85.000	0168/T	Turbo Tape	50.000
0177	Pilot (linguaggio)	70.000	0178	Ultra Basic	125.000
0179	Comp/Scompactor	50.000	0193/C	Basic 4.0	80.000
0194	Sprite Generator	70.000	0195	Assembler	80.000
0196	S.A.M. (Tronix)	100.000	0197	G-Pascal	95.000
0198	Forth 64 (Commodore)	95.000	0199	Tol 64	85.000
0200	Master (Commodore)	145.000	0201/C	Scheda CP/M (Commodore)	125.000
0333	Clone machine	100.000	0334	Unguard	200.000
0335	Fast Copy 4.5	100.000	0336	Music Composer	150.000
0337	Basic Programm Generator	250.000	0338	Copy Files	80.000
0320	The MANAGER 64	100.000	0321	Austro Compilatore	80.000
0210	Lite Pen (incluso software)	130.000			
		GESTIONE DATI			
0157	Easy Calc	125.000	0205	Super Base	175.000
0206	Magic Desk (Commodore)	75.000	0207	Koala Joystick	100.000
0209	Data Log.	120.000	0220	Easy Calc Tape	125.000
0400	Stock Control	120.000	0401	Easy Label	120.000

N.B. (*) Per configurazione base si intende Commodore 64, Floppy 1541, Stampante Commodore a 80 colonne.
I codici barrati 'T' sono disponibili anche su cassetta, quelli barrati 'C' sono Cartridge.

Leoni informatica

Direzione Commerciale • via Valsolda, 21
20143 MILANO - tel. 02/8467378

Negozio vendite al pubblico: Milano - Corso di Porta Romana, 123



IL LIBRO PIU' STRAORDINARIO PER UNO STRAORDINARIO COMPUTER

120 IDEE SU 63 ARGOMENTI

Animazioni - Architetti - Arredamento - Automobili - Banche - Barche - Brevetti - Business grafica - Carte intestate - Cartelli indicatori - Cataloghi - Cinema - Circuiti integrati - Convegni/Corsi - Descrizioni - Design - Dentisti - Disegnatori - Economia/Finanza - Edilizia - Editoria - Elettricisti - Elettronica - Enigmistica - Esercito - Falegname - Fonts - Fumetti - Giardinaggio - Gioielli - Graffiti - Grafici - Identikit - Informatica - Immagini grafiche - Ingegneria - Linee aeree - Marchi industriali - Mobili - Moda - Modellistica - Mostre/Saloni - Moto - Musica - Negozi - New York - Organigrammi - Orologi - Piante stradali - Pittura - Pubblicità - Regali - Robot - Scacchi - Scultura - Scuola - Simboli - Simmetrie - Space shuttle - Topografia - Urbanistica - Videogiochi - Vignette.

GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
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
	402 D	L. 30.000	

☐ 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 _____

n° _____

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ 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

Titolazione con lo Spectrum

$H^+ = \sqrt{K_1 \cdot C}$	pH iniziale.
$H^+ = ((100 - 2V)/(2\ V)) \cdot K_1$	pH sino a 50 ml non comprensi.
$H^+ = \sqrt{K_1 \cdot K_2}$	pH a 50 ml (I punto di equivalenza).
$H^+ = ((100 - v)/(v - 50)) \cdot K_2$	pH tra 51 e 99 ml.
$H^+ = \sqrt{(1e - 14 \cdot C)/K_2}$	pH a 100 ml (II punto di equivalenza).

K_1 = Costante di dissociazione primo idrogenione.
K_2 = Costante di dissociazione del secondo idrogenione.
C = Concentrazione dell'acido e della base.
V = Volume della base.

Figura 2. *Algoritmi per la titolazione dell'acido diprotico.*

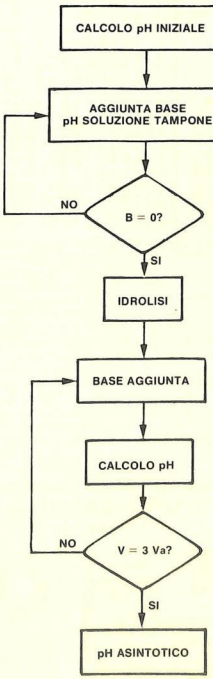


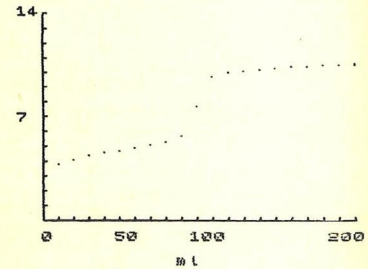
Figura 3. *Flow-chart per la simulazione della titolazione.*

é in edicola
VIDEO GIOCHI
é in edicola
Una pubblicazione
GRUPPO EDITORIALE JACKSON

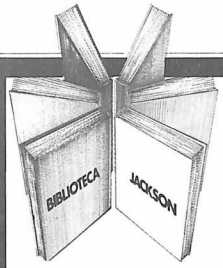
é in edicola
HOME COMPUTER
HC
UNA PUBBLICAZIONE FIRMATA
GRUPPO EDITORIALE JACKSON

Figura 4. *Un esempio di esecuzione.*

```
Titolazione acido debole-base
          forte

C.acido=.001
Vol.acido=100
Ka=.000018
C.base=.001
Var.volume=10
Analisi o grafico(a/g)?
pH iniziale=3.67
Volume=10    pH tampone=3.79
Volume=20    pH tampone=4.14
Volume=30    pH tampone=4.37
Volume=40    pH tampone=4.56
Volume=50    pH tampone=4.74
Volume=60    pH tampone=4.92
Volume=70    pH tampone=5.11
Volume=80    pH tampone=5.34
Volume=90    pH tampone=5.69
Volume=100   pH idrolisi=8.44
Vol.base in eccesso:10   pH=9.67
Vol.base in eccesso:20   pH=9.95
Vol.base in eccesso:30   pH=10.11
Vol.base in eccesso:40   pH=10.22
Vol.base in eccesso:50   pH=10.3
Vol.base in eccesso:60   pH=10.36
Vol.base in eccesso:70   pH=10.41
Vol.base in eccesso:80   pH=10.45
Vol.base in eccesso:90   pH=10.49
Vol.base in eccesso:100   pH=10.5
2
pH asintotico=11


Titolazione acido debole-base

C.acido=0
Vol.acido=0
Ka=0
C.base=0
Var.volume=0

Analisi o grafico(a/g)?
C.acido=.01
Ka=.000018
V.acido=100
C.base=.01
V.base=30

Tampone a pH=4.37

C.acido=.01
Ka=.000018
V.acido=100
C.base=.01
V.base=100


Idrolisi:pH=8.37


Esempio di titolazione di acido
        diprotico
        (100 ml)

K1=.0013
K2=1.5E-10
C.acido=.001
V.base=10
grafico(s/n)?
pH iniziale=2.94
Vol.base=10
pH =2.28
Vol.base=20
pH =2.7
Vol.base=30
pH =3.06
Vol.base=40
pH =3.48
Vol.base=50
I punto di equivalenza:pH=6.35
V.base=60
pH=9.22
V.base=70
pH=9.64
V.base=80
pH=10
V.base=90
pH=10.42
V.base=100
II punto di equivalenza:pH=10.41


Titolazione acido debole-base

C.acido=.01
Vol.acido=100
Ka=.000018
C.base=.01
Var.volume=10
Analisi o grafico(a/g)?
pH iniziale=3.37
Volume=10    pH tampone=3.79
Volume=20    pH tampone=4.14
Volume=30    pH tampone=4.37
Volume=40    pH tampone=4.56
Volume=50    pH tampone=4.74
Volume=60    pH tampone=4.92
Volume=70    pH tampone=5.11
Volume=80    pH tampone=5.34
Volume=90    pH tampone=5.69
Volume=100   pH idrolisi=9.44
Vol.base in eccesso:10   pH=10.67
Vol.base in eccesso:20   pH=10.95
Vol.base in eccesso:30   pH=11.11
Vol.base in eccesso:40   pH=11.22
Vol.base in eccesso:50   pH=11.3
Vol.base in eccesso:60   pH=11.36
Vol.base in eccesso:70   pH=11.41
Vol.base in eccesso:80   pH=11.45
Vol.base in eccesso:90   pH=11.49
Vol.base in eccesso:100   pH=11.5
2
pH asintotico=12
```




```
C.acido=.001
Ka=.000018
V.acido=100
C.base=.002
V.base=45

Tampone a pH=5.69

C.acido=.001
Ka=.000018
V.acido=50
C.base=.002
V.base=60
pH=10.8
```



Libri firmati JACKSON

Pierre Lotigie-Laurent
UN PERSONAL COMPUTER FIRMATO IBM
Il libro offre una panoramica completa sul Personal Computer IBM. Comincia col descrivere l'hardware, passa poi alle due applicazioni, alla programmazione e ai sistemi operativi.
Illustra le diverse configurazioni che si possono realizzare per soddisfare esigenze di utenti diversi, e finisce con un breve manuale per l'installazione e l'uso.
Una preziosa prefazione all'Edizione Italiana integra ed aggiorna l'edizione originale.
L'immagine che ne esce, grazie allo stile e al linguaggio, è quella di un'informatica abbordabile e cordiale, pronta a venire in aiuto sul lavoro e a istruire e divertire in famiglia.
152 pagine.
Codice **404H** **L. 15.000**

Stanley R. Trost
PROGRAMMI UTILI PER IBM PC
Il libro presenta una serie di oltre 65 programmi per il Personal Computer IBM, ampiamente collaudati e pronti per l'uso.
Dopo una breve introduzione, che vi metterà in grado di usare i programmi proposti, seguono i programmi stessi suddivisi per capitoli: finanza e bilancio familiare, gestione commerciale, beni immobili, analisi di dati, registrazione di dati, esercizi di matematica.
I programmi sono da "usare", e il libro non ha la pretesa di insegnarvi la programmazione in , BASIC, ma proprio l'uso dei programmi vi porterà gradualmente, e quasi automaticamente, ad apprenderne le tecniche fondamentali; può essere un buon punto di partenza per diventare voi stessi programmatori.
E proprio per aiutarvi in questo, nelle Appendici è presentata una serie di sottoprogrammi di uso generale, usati dai programmi del libro, e vi viene spiegato con facili esempi come inserirli nei vostri nuovi programmi per risparmiare tempo e fatica.
180 pagine.
Codice **564 D** **L. 15.000**

La Biblioteca che fa testo

GRUPPO EDITORIALE JACKSON

Attenzione compilare per intero la cedola
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. 3000 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

centrazione noti (righe 390-411);
8 - calcolo del pH di una soluzione di acido diprotico debole con una base forte durante una titolazione (righe 500-730).

Le formule usate per le opzioni 1-5 sono facilmente reperibili su qualsiasi testo di chimica, mentre per le rimanenti una tabella a parte illustrerà gli algoritmi usati.
Per il calcolo del pH di acidi e basi forti, c'è un controllo per verificare se la concentrazione è diluita (meno di 10^{-6} M).
Se la soluzione è diluita, viene applicata una formula correttiva che tiene conto degli idrogenioni dell'acqua ed impedisce grossolani errori (pH 8 per una soluzione di HCl 10^{-8} M).
Per il pH della titolazione, una tabella illustra il relativo flow-chart.
Si noti che quando il volume della base aggiunta, supera di tre volte il volume iniziale dell'acido, il pH, ormai asintotico, viene calcolato nella riga 390.
Per quanto riguarda la titolazione dell'acido diprotico, per semplicità si è assunto che le concentrazioni dell'acido e della base fossero uguali ed inoltre il volume di acido da titolare fisso (100 ml).
La grafica è molto semplice. Le righe 900-905 costituiscono la subroutine che disegna gli assi e le divisioni della scala di riferimento con le relative scritte.

Le righe 950-1090 ricalcolano i vari valori di pH durante la titolazione e li plottano sul grafico. Analogamente le righe 799-885 provvedono a tracciare i punti della titolazione dell'acido diprotico.
Non si sono utilizzate delle subroutine che calcolassero una volta sola i valori di pH della titolazione sia per i grafici che per la tabulazione dei valori stessi, perché sarebbero risultate molto più complicate e più lunghe della semplice ripetizione degli algoritmi e per i grafici e per le tabulazioni. L'esecuzione del programma è semplicissima: basta dare il RUN e seguire le indicazioni del programma immettendo i dati richiesti. Alla fine del calcolo richiesto, il programma o si ferma da solo, oppure si preme la lettera "f"; ciò è fatto perché in alcune opzioni (ad esempio la 7) può essere comodo impostare continuamente altri dati e fermare il tutto alla fine della ricerca.

Un'ultima avvertenza: è bene che la concentrazione dell'acido da titolare non sia troppo bassa (l'optimum è tra $10^{-1} \div 10^{-3}$ M) perché altrimenti al pH iniziale e nei punti di equivalenza c'è interferenza degli idrogenioni dell'acqua. In questi casi bisogna introdurre algoritmi di calcolo alquanto più complicati, che rientrano già nel lavoro più specialistico. ■

NEL PROSSIMO NUMERO DI

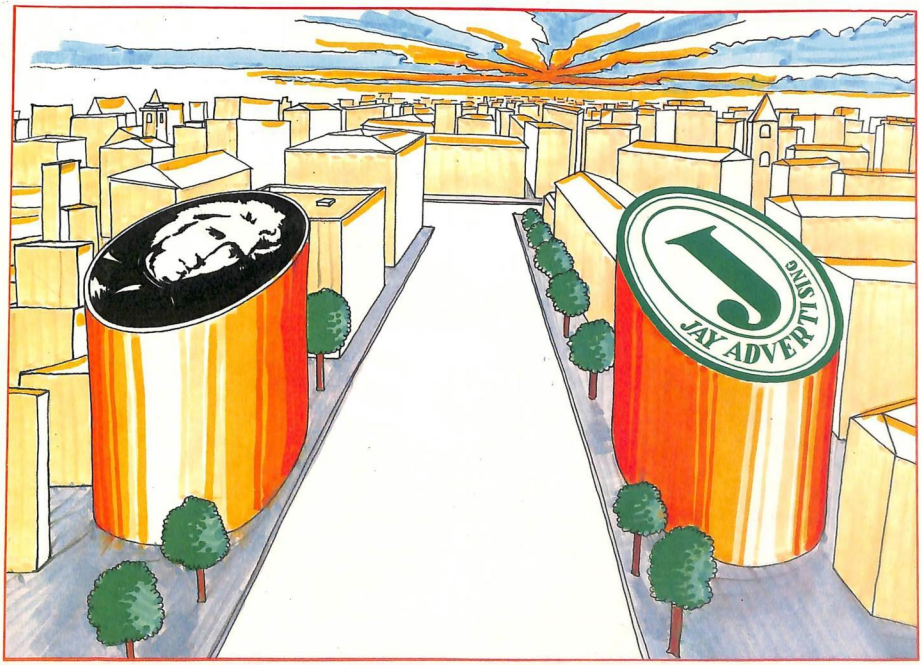
PERSONAL SOFTWARE

TROVERETE

- SUPER CONVERTER PER SPECTRUM
- DAMA PER ZX81
- ORGANO ELETTRONICO CON LO SHARP
- LANGUAGE TUTOR PER SPECTRUM
- APPLE: PAGINE VIDEO E DINTORNI
- SPRITE EDITOR PER C 64
- IL CASTELLO PER SPECTRUM
- SUPER ASSEMBLER PER C 64

DAL 1° DICEMBRE 1984

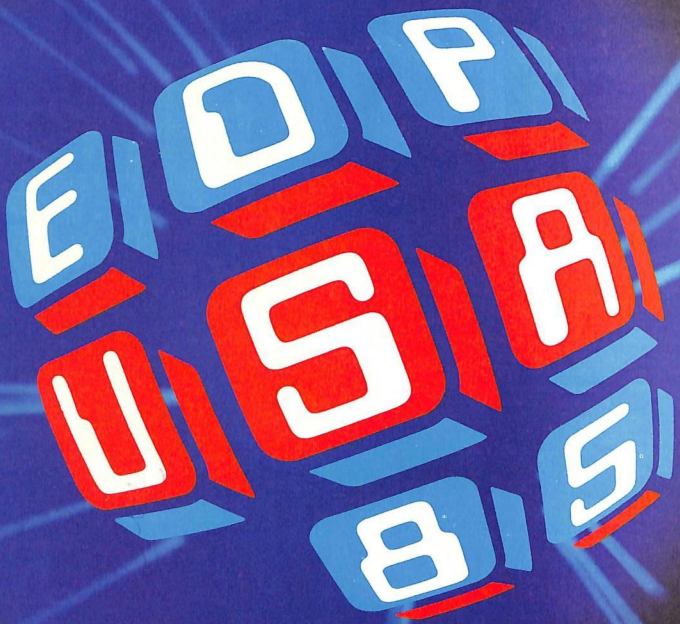
J. ADVERTISING E JACKSON PIU' VICINI!



Dal 1° dicembre 1984 J. ADVERTISING, concessionaria esclusiva di pubblicità per tutte le riviste del Gruppo Editoriale Jackson, Computer Publishing Group e J. Soft ha cambiato indirizzo. Il nuovo indirizzo è:

J. ADVERTISING s.r.l.
V.le Restelli, 5
20124 MILANO
Tel. 02/6882895-6882458-6880606

MILANO
5~8 FEBBRAIO
1985 MOSTRA E SEMINARIO



COMPUTER GRAPHICS

Tutte le più sofisticate proposte della tecnologia americana nel settore dell'informatica "concentrate" in un'unica mostra, una mostra che specializzandosi ulteriormente di anno in anno, è giunta alla XIV edizione.
Riproponendo un'iniziativa che già lo scorso anno è stata accolta con grande interesse dagli operatori del settore, un intero padiglione verrà dedicato esclusivamente al COMPUTER GRAPHICS ed alle sue applicazioni.
In concomitanza con la mostra e quale momento di aggiornamento e approfondimento delle più attuali tematiche dell'informatica, si terranno due seminari di studio:
6-7 FEBBRAIO - in collaborazione con la CITIBANK N.A.: "Cinque tecnologie innovative per l'informatica e loro applicazioni: integrazione circuitale, optical disc, I/O voice, business graphics e linguaggi della quarta generazione".
Coordinatore: Dr. Gianfranco Minati.
Quota di partecipazione:
Lire 200.000.
8 FEBBRAIO: "La grafica nella realtà aziendale e industriale".
Coordinatore: Ing. Roberto Favero.
Quota di partecipazione:
Lire 100.000.
Per ulteriori informazioni sulla mostra e sulle modalità di partecipazione al seminario, contattare: CENTRO COMMERCIALE AMERICANO - TEL. 02/4696451 - TELEX 330208 USIMC I.

USA

CENTRO COMMERCIALE AMERICANO

Via Gattamelata 5 - 20149 Milano
Tel. 02/4696451 Telex 330208 USIMC I

PERSONAL SOFTWARE

Word processor per TI99/4A

Per studiare con il computer e immagazzinare tutto su nastro magnetico

di *Stefano di Tommaso*

Introduzione

Cosa significa possedere un personal computer? Molto spesso significa utilizzarlo per mille cose che altrimenti sarebbero state fatte con procedure tradizionali.
Tra queste sicuramente l'elementare operazione del prendere carta e matita e scrivere degli appunti. Ma per far ciò con il computer bisogna che in memoria vi siano delle istruzioni che ordinano al computer di ricevere gli appunti sotto forma di dati, di catalogarli e di trasferirli in memoria.
Da queste elementari considerazioni hanno avuto l'avvio i primi programmi di trattamento dei testi, cioé di word processing, con l'aggiunta però di numerose sofisticazioni in ordine alla possibilità di maneggiare i testi stessi, modificarli in tutto o in parte, ricomporli in ordine eccetera.
Presentiamo in questo numero finalmente qualcosa di simile anche per i lettori che dispongono del Texas TI99/4A, spiegandone inoltre il funzionamento ed il modo in cui è strutturato.

Cosa può fare

È un programma che vi permetterà di scrivere tutto quello che volete sul video del vostro computer sfruttandone le buone capacità di editing per spostare le parole FCTN (2)+SPACE BAR; cancellarle FCTN (1); inserirle FCTN (2) + PAROLA, scrivendo riga per riga, pagina per pagina, per poi trasferire le pagine su cassetta in maniera abbastanza efficiente.
Inoltre potrete rileggere tutto quanto scritto, oppure solo una parte,

Listato 1. *Il programma Word processor.*

```
???
1000 CALL CLEAR
1010 CALL SCREEN(5)
1020 CALL COLOR(13,3,3)
1030 FOR I=1 TO 170
1040 READ V,H
1050 CALL VCHAR(V,H,128,1)
1060 DATA 4,7,4,11,4,13,4,14,4,1
5,,16,4,17,4,19,4,20,4,21,4,22,
4,24,4,25,4,26
1070 DATA 5,7,5,11,5,13,5,17,5,1
9,5,22,5,24,5,27
1080 DATA 6,7,6,11,6,13,6,17,6,1
9,6,22,6,24,6,27
1090 DATA 7,7,7,9,7,11,7,13,7,17
,7,19,7,21,7,24,7,27,8,8,8,10,8,
13,8,14,8,15
1100 DATA 8,16,8,17,8,19,8,22
1110 DATA 8,24,8,25,8,26
1120 DATA 10,3,10,4,10,5,10,6,10
,8,10,9,10,10,10,11,10,13,10,14,
10,15,10,16,10,18,10,19,10,20,10
,21
1130 DATA 10,23,10,24,10,25,10,2
6,11,3,11,6,11,8,11,11,11,13,11,
16,11,18,11,23,12,3,12,4,12,5,12
,6
1140 DATA 12,8,12,11,12,13,12,16
,12,18,12,23,12,24,12,25,12,28,1
2,29,12,30,13,3,13,8,13,10
1150 DATA 13,13,13,16,13,18,13,2
3,14,3,14,8,14,11,14,13,14,14,14
,15,14,16,14,18,14,19,14,20,14,2
1
1160 DATA 14,23,14,24,14,25,14,2
6
1170 DATA 16,3,16,4,16,5,16,6,16
,8,16,9,16,10,16,11,16,13,16,14,
16,15,16,16,16,18,16,19,16,20
1180 DATA 16,21,17,3,17,8,17,13,
17,16,17,18,17,21,18,3,18,4,18,5
,14,6,18,8,18,9,18,10,18,11,18,1
3
1190 DATA 18,16,18,18,18,21,19,6
,19,11,19,13,19,16,19,18,19,20,2
0,3,20,4,20,5,20,6,20,8,20,9,20,
10
1200 DATA 20,11,20,13,20,14,20,1
5,20,16,20,18,20,21
1210 NEXT I
1220 CALL COLOR(13,5,5)
1230 FOR COL=1 TO 12
```

Seguito listato Word processor.

```
1240 CALL COLOR(COL,2,10)
1250 NEXT COL
1260 CALL CHAR(128,"FFFFFFFFFFFF
FFFF")
1270 GOSUB 2740
1280 PRINT "     ___________________
.____
         WORD    PROCESSOR
                              ": :
1290 PRINT "            OPZIONI:
     ___________________________
__": : : : :
1300 PRINT "           premi": :
:
1310 PRINT "(1)se vuoi leggere i
 files  (2)se vuoi scrivere pagi
ne  (3)se  hai finito
(4)se vuoi registrare i dati"
1320 PRINT "(5)se vuoi rileggere
  pagine(6)se vuoi aggiornare pa
gine(7)se vuoi cancellare tutto"

1330 R=0
1340 CALL KEY(0,R,ST)
1350 IF ST=0 THEN 1340
1360 IF (R<49)+(R>55)THEN 1340
```

```
1370 ON R-48 GO SUB 1940,1410,13
90,2110,2260,2450,2650
1380 GOTO 1270
1390 DISPLAY : : : : : : : : : :
 : : :"     CIAOOOOOO......": :
: : : : : : :
1400 END
1410 DIM A$(23,10)
1420 GOSUB 2740
1430 INPUT "quante pagine al mas
simo ?":N
1440 IF N>10 THEN 1430
1450 INPUT "da quale  pagina  sc
rivi ?":PP
1460 IF PP>N THEN 1450
1470 GOSUB 2740
1480 FOR C=PP TO N
1490 PRINT : : :TAB(9);STR$(C)&"
^a  pagina": : :
1500 FOR I=RI+1 TO 23
1510 INPUT A$(I,C)
1520 IF LEN(A$(I,C))>28 THEN 153
0 >LSE 1550
1530 A$(I,C)=SEG$(A$(I,C),1,28)

1540 PRINT A$(I,C)
1550 CALL KEY(0,RED,STRED)
1560 IF STRED=0 THEN 1550
```

modificare, aggiornare e cancellare tutte le pagine.
Se si dispone di una stampante basterà l'aggiunta di poche istruzioni per averne una copia su supporto cartaceo.
Il programma comunque è riferito alla configurazione di sistema più diffusa da noi, cioé: console, televisore, registratore a cassette.

Con questo tipo di applicazione è possibile sfruttare un vantaggio essenziale del Texas TI99/4A nei confronti di molti suoi concorrenti: quello di poter usufruire di una comoda tastiera di tipo professionale. Praticamente dunque sarà come battere a macchina, con in più la possibilità di cancellare gli errori, scrivere ordinatamente, trasferire le pagine scritte su cassetta in pochi minuti di registrazione e... "sigillarle" con un proprio codice segreto. Infatti bisognerà conoscere sia la parola (o le parole)-chiave, per accedere alla lettura del nastro magnetico, sia il numero del file sul quale si è trasferito il testo scritto, altrimenti il programma si ferma. Da notare è che la parola-chiave non è scritta nel listato, ma sul nastro stesso, e non si potrà mai leggerla: bisogna conoscerla.
Data la limitata estensione della memoria RAM del computer non è possibile inserire contemporaneamente in memoria più di 10 pagine dattiloscritte. Ma ciò non costituisce un reale limite all'operatività del programma poiché si possono caricare i dati sul supporto magnetico e continuare nella scrittura finché si vuole.
Su una cassetta di tipo C 60 sarà possibile inserire fino a 70 pagine dattiloscritte. Con un buon risparmio d'ingombro, dunque.

Come funziona il programma

Il programma si apre con il titolo word processor scritto su tutta l'ampiezza del video, mentre si compone lentamente, riga per riga, per poi cancellarsi con lo scrolling orizzontale, da sinistra verso destra.
A questo punto compare il menù d'apertura, riportato nella figura 1.
Ai punti (1) e (4) ci si rivolge direttamente all'utilizzo dei file, rispettivamente, per estrarli dal nastro, e per scriverli. Non prima però di aver scritto la parola segreta ed il numero del file, separati da una virgola. Al punto (2) si va a scrivere nuove pagine, scegliendo il numero totale delle pagine da comporre, e la prima pagina dalla quale scrivere. Per esempio si può scegliere di scrivere su 8 pagine a partire però dalla 4 pagina, se le precedenti 3 erano già state scritte.
Si scrive di riga in riga (figura 2) componendo le parole fino all'esatta lunghezza di ogni riga, con 28

WORD PROCESSOR

OPZIONI

Premi
(1) Se vuoi leggere il file
(2) Se vuoi scrivere pagine
(3) Se vuoi finire
(4) Se vuoi registrare i dati
(5) Se vuoi rileggere pagine
(6) Se vuoi aggiornare pagine
(7) Se vuoi cancellare tutto

Figura 1. *Il menu d'apertura.*

Libri firmati JACKSON

Roberto Rigo
SPECTRUM TOOL
Programmi di utilità, grafica e gioco

Una giusta via di mezzo tra la praticità di uso e la velocità di esecuzione di un programma è rappresentata dall'uso combinato del BASIC e del linguaggio macchina; la conoscenza di quest'ultimo è comunque indispensabile per poter sfruttare a pieno il proprio calcolatore.
Questa è la premessa da cui parte l'autore, per proporre una serie di interessanti programmi BASIC che si servono di routine scritte in linguaggio macchina.

182 pagine
Codice **554D L. 15.000**

Joseph Kascmer
FACILE GUIDA AL COMMODORE 64

Questo libro vi insegnerà in poche ore ad usare il vostro Commodore 64, cominciando dalla tastiera e dal video, per passare poi alle altre periferiche più comuni, l'unità a dischi e il registratore a cassette.
Imparerete con estrema semplicità a scrivere programmi in BASIC, ma se questo non è il vostro obiettivo potete "saltare" i Capitoli dedicati alla programmazione e imparare invece come utilizzare "pacchetti" di software preconfezionati.

140 pagine
Codice **400D L. 13.500**

Francesco Lentini
CONOSCI TE STESSO TRAMITE IL TUO PERSONAL COMPUTER

Il libro si compone di cinque capitoli, ognuno dei quali attinge ad una delle discipline che affrontano in qualche modo lo studio della personalità umana: Numerologia, Astrologia, Grafologia, Scienza del Comportamento, Test di Intelligenza.
I programmi sono stati scritti per Apple II e sono disponibili su cassetta e dischetto; nelle due Appendici ci sono i consigli necessari per adattarli a qualsiasi altro PC della stessa generazione.

128 pagine
Codice **401D L. 12.000**

La Biblioteca che fa testo

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 _____ n° _____

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ 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

GRUPPO EDITORIALE JACKSON

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

Il Jacksoniano ha il Basic

Video Basic, corso su cassetta per parlare subito

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 dialogare col tuo computer, sia un VIC 20, un Commodore 64 o un Sinclair Video Basic è in edicola. Provalo subito.

Oggi il Basic si impara così.

GRUPPO ETHOS

Facile

in mano.

col tuo computer.

GRUPPO EDITORIALE JACKSON

IN EDICOLA DALL' 8·1·'85

In omaggio
una fantastica cassetta giochi.

3ª Pagina

Caro Ernesto,
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla

Figura 2. *La scrittura avviene di riga in riga.*

caratteri disponibili; se si va oltre questi, quando premete ENTER appare la riga come l'avete composta, ma tagliata oltre il 28° carattere. Se a questo punto ciò vi sta bene, premete ancora ENTER e passate alla riga successiva, altrimenti potete scegliere fra una serie di operazioni alternative corrispondenti ad altrettanti tasti-funzione:

- FCTN (8) (=BACK) serve per scrivere da capo quella stessa riga;
- FCTN (9) (=REDO) serve per rileggere le precedenti righe della pagina sulla quale state scrivendo, già ricomposte, per proseguire poi nella scrittura;
- FCTN (6) (=PROC'D) serve per proseguire con la pagina successiva, interrompendo a quel punto la pagina che state scrivendo non prima però di averla riletta già incolonnata, e aver premuto un tasto dopo di ciò.

Quando premete PROC'D dopo la lettura della pagina appare una serie di opzioni, per scrivere da capo, per proseguire con la pagina successiva o infine per tornare al quadro principale.

Continuando a scorrere il menu principale, vediamo che al punto (3) il programma si fermerà, al punto (7) cancellerà tutto ciò che è stato scritto, al punto (5) sarà possibile leggere una o tutte le pagine. In quest'ultimo caso si prema un tasto dopo aver letto ogni pagina. Infine al punto (6) potrete aggiornare una o più pagine, inserendo il numero della pagina e quello della riga da cambiare.

Le righe di ogni pagina sono 23 per cui per ogni numero superiore che inserite dovrete rifarlo da capo. Se volete aggiungere soltanto delle righe, basterà iniziare a scrivere dalla riga che avete inserito in avanti, fino alla fine della pagina, ma se usate il comando PROC'D allora avete finito di correggere.

È importante da ultimo ricordare che, quando stabilite il numero massimo di pagine da compilare, questo numero resterà tale anche nell'operazione di trasferimento dei dati su cassetta, e sarà inserito nel file. Così se avete scritto 4 pagine ma avevate posto un massimo di 6 pagine, il registratore scriverà e leggerà dal file sempre 6 pagine, comprese quelle vuote.

Seguito listato Word processor.

```
1570 IF RED=6 THEN 1510
1580 REM
1590 IF RED=12 THEN 1660
1600 READING=99
1610 IF RED=15 THEN 1670
1620 READING=0
1630 NEXT I
1640 IF MODYFY=99 THEN 2550
1650 GOTO 1910
1660 GOSUB 2740
1670 FOR II=1 TO 23
1680 IF READING=99 THEN 1690 ELS
E 1700
1690 IF A$(II,C)="" THEN 1710
1700 PRINT A$(II,C)
1710 NEXT II
1720 IF READING=99 THEN 1730 ELS
E 1740
1730 GOTO 1620
1740 CALL KEY(0,LETTURA,STLET)
1750 IF STLET=0 THEN 1740
1760 GOSUB 2740
1770 IF MODYFY=99 THEN 2550
1780 PRINT "          opzioni":
: :
```

```
1790 IF C>=N THEN 1810
1800 PRINT "questa era la "&STR$
(C)&"^ pagina su";:N;"pagine  a
 disposizione"
1810 PRINT "(1)per rifare ultima
  pagina(2)per fare  prossima pa
gina(3)per ritornare al quadro
   principale"
1820 RIT=0
1830 CALL KEY(0,RIT,ST)
1840 IF ST=0 THEN 1830
1850 IF (RIT>51)+(RIT<49)THEN 18
30
1860 ON RIT-48 GOTO 1490,1880,18
70
1870 RETURN
1880 IF C>=N THEN 1890 ELSE 1910

1890 PRINT "          ATTENZIONE
        NON PUOI CONTINUARE"

1900 GOTO 1810
1910 NEXT C
1920 GOTO 1760
1930 REM
1940 GOSUB 2740
1950 INPUT "chiave , numero file
```

VIDEO BASIC abbonarsi conviene

(5 splendidi raccoglitori insieme al corso completo)

Video Basic lo trovi in edicola a lire 8.000 il fascicolo con cassetta e manuale. Ma abbonarsi conviene; con 165.000 lire avrai infatti il corso completo, a casa tua, e 5 splendidi (e pratici) raccoglitori del valore di 40.000 lire.
NON PERDERE L'OCCASIONE!

Desidero abbonarmi a Video Basic

- ☐ Per il computer Commodore VIC 20
- ☐ Per il computer Commodore 64
- ☐ Per il computer Sinclair Spectrum

Spedire a:
JACKSON
Via Rosellini, 12
20124 Milano

Allego lire 165.000 con assegno n° _____ della Banca _____ o allego fotocopia della ricevuta di versamento con vaglia postale intestato a GRUPPO EDITORIALE JACKSON - MILANO, che mi dà diritto di ricevere a casa mia il corso completo e 5 raccoglitori.

Nome _____ Cognome _____

Via _____ N. _____

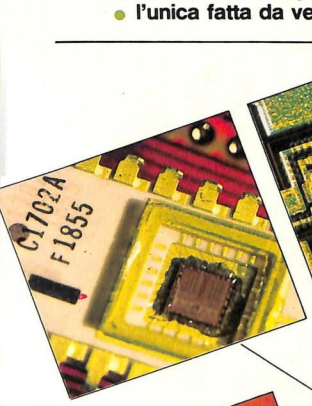
CAP _____ Città _____ Provincia _____

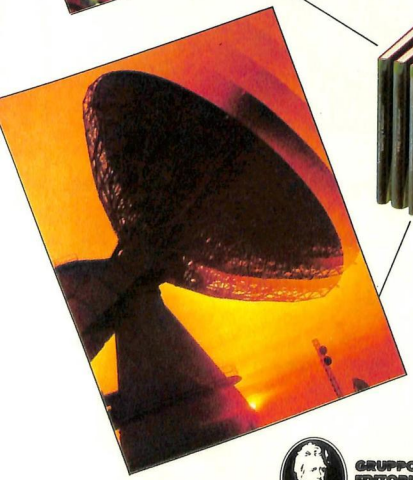
TUTTO SU...

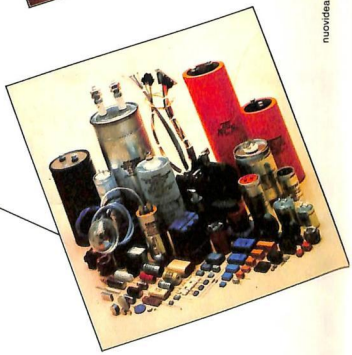
Un'opera prestigiosa

Elettronica – Informatica – Comunicazioni

- l'unica cosi completa e organica
- l'unica fatta da veri specialisti
- l'unica che non sarà mai sorpassata dai fatti







nuovidea

GRUPPO EDITORIALE JACKSON

In collaborazione con il Learning Center
TEXAS INSTRUMENTS

OGNI SETTIMANA IN EDICOLA 2 DISPENSE

FA SCUOLA

Enciclopedia di Elettronica e Informatica

✓ I temi affascinanti della civiltà del computer, gli sviluppi della società tecnologica in un opera creata per capire e affrontare il micromillennio.

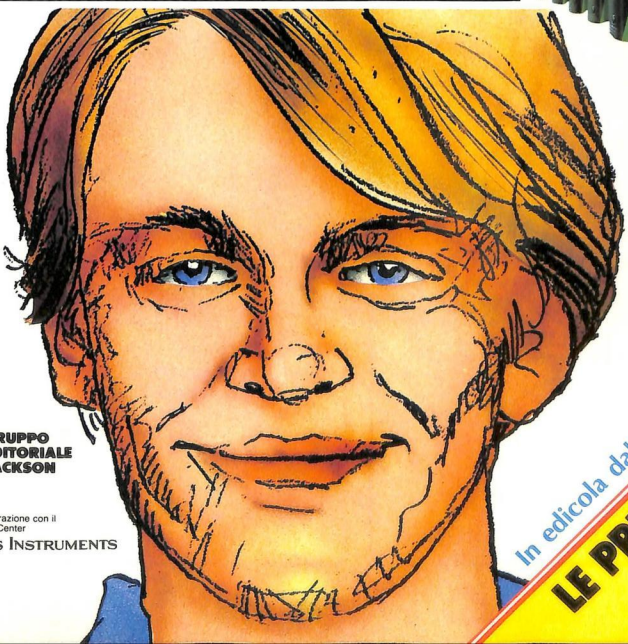
✓ **L'enciclopedia giovane e pratica,** che nasce dai progressi della ricerca, che parla il linguaggio chiaro e conciso della "bit generation".

✓ **Lo strumento base per chi studia,** per chi lavora, per chi vuol vivere da protagonista le affascinanti novità del nostro tempo e prepararsi a quelle del futuro prossimo venturo.

In edicola
60 dispense
30 appuntamenti settimanali con gli esperti JACKSON e i tecnici TEXAS INSTRUMENTS

✓ **Ogni settimana:**
56 pagine di elettronica, informatica e comunicazioni.

✓ **In sole 30 settimane**
una splendida opera per la vostra biblioteca:
7 prestigiosi volumi
1600 pagine complessive
700 foto e 2200 disegni a colori



GRUPPO EDITORIALE JACKSON

In collaborazione con il Learning Center
TEXAS INSTRUMENTS

In edicola dal 13 Novembre 1984

LE PRIME 2 DISPENSE SOLO LIRE 2.500

Con un'inutile perdita di tempo. Per evitarla potrete andare al punto (6) del menu (Aggiornamento) e reinserire il nuovo numero di pagine senza cambiare alcunché.

Come è strutturato il programma

L'immagine iniziale è stata ottenuta creando un carattere pieno (quadratino), e disponendolo numerose volte (170) sullo schermo mediante l'istruzione CALL HCHAR fino a formare l'immagine voluta. Con le istruzioni READ ... DATA il Computer legge ogni volta le coordinate di ogni punto.
Dopodiché si giunge al menu d'apertura che divide il programma in 7 subroutine principali, alcune delle quali strutturate.
Molto spesso esse si fondono tra di loro, questo per utilizzare più volte alcune istruzioni in comune. Quando ciò avviene vengono usate delle variabili "fittizie" (Reading, Modify) che assumono valore = 99 oppure 0, creando così la possibilità di non confondere le due diverse sequenze che in quel punto si incrociano.
I testi, mentre sono scritti, vengono trasferiti nel buffer input/output tramite l'istruzione di INPUT, perciò ogni volta che scrivete una riga state attribuendo una serie di valori alla variabile di stringa A$ tramite l'istruzione di INPUT.
Ecco perché alla fine di ogni riga bisognerà premere ENTER.
Per catalogare queste variabili di stringa abbiamo usato una matrice di (23x10) punti corrispondenti a 23 righe per 10 pagine.
L'istruzione in BASIC è DIM A$ (23,10).
I "puntatori" di questa matrice a due dimensioni sono le variabili (I) e (C). Per utilizzarli abbiamo costruito due cicli nidificati (figura 3).
In questo modo è possibile passare per ogni punto della matrice e compiere le operazioni desiderate.
Inoltre, quando abbiamo affrontato il problema dell'immagazzinamento

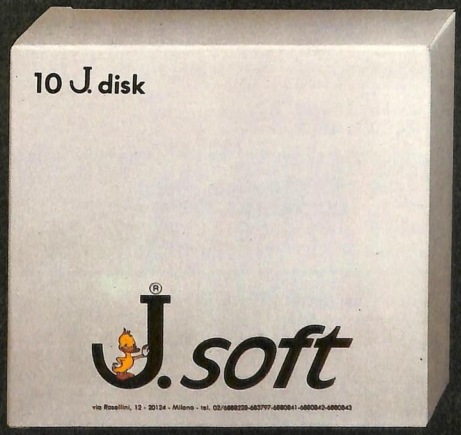
```
FOR C = 1 TO 10
        Righe di programma
FOR I = 1 TO 23
        Righe di programma
INPUT A$ (I,C)
        Righe di programma
NEXT I
        Righe di programma
NEXT C
```

Figura 3. *I due cicli nidificati per utilizzare i puntatori di DIM A$.*

Seguito listato Word processor.

```
 ?":Q$,Q
1960 OPEN #Q:"CS1",INPUT ,INTERN
AL,FIXED 192,SEQUENTIAL
1970 INPUT #Q:N,QQ$
1980 IF QQ$=Q$ THEN 1990 ELSE 19
70
1990 FOR C=1 TO N
2000 PRINT : :"STO LEGGENDO LA "
;C;"PAGINA"
2010 FOR I=1 TO 20 STEP 5
2020 PRINT "STO LEGGENDO LA ";:I
:I+1:I+2:I+3:I+4;" RIGA"
2030 INPUT #Q:A$(I,C),A$(I+1,C),
A$(I+2,C),A$(I+3,C),A$(I+4,C)
2040 NEXT I
2050 PRINT "STO LEGGENDO LA ":"
21":" 22":" 23 RIGA "
2060 INPUT #Q:A$(21,C),A$(22,C),
A$(23,C)
2070 NEXT C
2080 CLOSE #Q
2090 RETURN
2100 REM
2110 GOSUB 2740
2120 INPUT "chiave , numero file
 ?":QQ$,Q
2130 OPEN #Q:"CS1",OUTPUT,SEQUEN
TIAL,FIXED 192,INTERNAL
2140 PRINT #Q:N,QQ$
2150 FOR C=1 TO N
2160 PRINT : :"STO SCRIVENDO LA"
;C;"PAGINA"
2170 FOR I=1 TO 20 STEP 5
2180 PRINT "STO SCRIVENDO LA":;I
:I+1:I+2:I+3:I+4;"RIGA"
2190 PRINT #Q:A$(I,C),A$(I+1,C),
A$(I+2,C),A$(I+3,C),A$(I+4,C)
2200 NEXT I
2210 PRINT "STO SCRIVENDO LA":"
21":" 22":" 23 RIGA"
2220 PRINT #Q:A$(21,C),A$(22,C),
A$(23,C)
2230 NEXT C
2240 CLOSE #Q
2250 RETURN
2260 MODYFY=0
2270 GOSUB 2740
2280 PRINT "vuoi rileggere una o
 tutte le pagine (U/T)"
2290 UT=0
2300 CALL KEY(0,UT,STUT)
2310 IF STUT=0 THEN 2300
2320 IF (UT=85)+(UT=117)THEN 279
0
```

AFFIDA I TUOI DATI A UN SUPPORTO SICURO



Come editori di software, abbiamo sentito l'esigenza di utilizzare, per la produzione dei nostri programmi, un supporto particolarmente affidabile. Dopo severi ed accurati test abbiamo operato la scelta. Siamo lieti di proporlo con il nostro marchio a chiunque desideri lavorare con la nostra stessa tranquillità. Floppy disk da 5'' 1/4, singola faccia, doppia densità, in confezione da 10 dischetti. Ordine minimo 10 dischetti. Ordini superiori solo multipli di 10 secondo la seguente scala di prezzi

- 10 dischetti	L. 5.000 cad.
- da 20 a 50 dischetti	L. 4.700 cad.
- da 60 a 100 dischetti	L. 4.400 cad.
- da 110 dischetti e oltre	L. 3.900 cad.

I prezzi sono comprensivi di IVA e spese di spedizione.

Per ordinare ritagliate e spedite il tagliando sotto riportato a
J. *soft* - via Rosellini, 12 - 20124 Milano
Tel. 02/6888228 - 683797 - 6880841 - 6880842 - 6880843

CEDOLA DI ORDINAZIONE OFFERTA DISCHETTI

Da compilare e spedire in busta chiusa a
J. *soft* - via Rosellini, 12 - 20124 Milano
Tel. 02/6888228 - 683797 - 6880841 - 6880842 - 6880843

Ordino i seguenti dischetti, in confezione da 10 pezzi cad., per un importo totale di L. ..... IVA e spese di spedizione incluse.

- ☐ N. ..... dischetti (minimo 10 e multipli di 10)
- ☐ 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 dischetti

Nome .....
Cognome .....
Via .....
CAP ..... Città ..... Prov. .....
Se richiesta fattura - codice fiscale .....
Data ..... Firma .....

Offerta valida solo per l'Italia.

Word processor per TI99/4A

Seguito listato Word processor.

```
2330 IF (UT=84)+(UT=116)THEN 234
0 ELSE 2300
2340 FOR C=1 TO N
2350 FOR I=1 TO 23
2360 PRINT A$(I,C)
2370 NEXT I
2380 IF MODYFY=99 THEN 2520
2390 CALL KEY(0,RR,SS)
2400 IF SS=0 THEN 2390
2410 IF (UT=84)+(UT=116)THEN 243
0
2420 RETURN
2430 NEXT C
2440 RETURN
2450 REM
2460 PRINT : : : :"   reinserisc
i il numero    di  pagine  pr
eferito       (precedente = "&ST
R$(N)&" )";
2470 INPUT N
2480 INPUT "
       quale pagina vuoi aggior
nare":C
2490 IF C>10 THEN 2480
2500 MODYFY=99
2510 GOTO 2350
2520 INPUT "da quale riga ?":RI
2530 RI=RI-1
2540 GOTO 1500
2550 RI=0
2560 GOSUB 2740
2570 PRINT "            PREMI :":
: :
2580 PRINT "(1)per aggiornare an
cora    (2)per tornare al quadro
     principale"
2590 CA=0
2600 CALL KEY(0,CA,STCA)
2610 IF STCA=0 THEN 2600 ELSE 26
20
2620 IF CA-48=1 THEN 2480
2630 MODYFY=0
2640 RETURN
2650 REM
2660 FOR C=1 TO N
2670 FOR I=1 TO 23
2680 LET A$(I,C)=""
2690 PRINT "       CANCELLAZIONE
"
2700 NEXT I
2710 NEXT C
2720 N=0
2730 RETURN
2740 CALL VCHAR(1,1,32,32*24)
2750 CALL VCHAR(1,1,128,48)
2760 CALL VCHAR(1,31,128,48)
2770 RETURN
2780 REM
2790 INPUT "quale pagina vuoi ri
leggere":C
2800 GOSUB 2350
2810 GOTO 2440
```

dei testi nei file su cassetta, abbiamo trovato utilissimo il fatto che ogni riga fosse già "squadrata" e dimensionata al massimo su 28 caratteri, perché così abbiamo potuto stabilire in anticipo la lunghezza del Record logico (28 caratteri + 1 x 5 righe), ed adattarlo su un record fisico che lo contenesse ("Fixed 192"), variando, al variare della lunghezza del testo, solo il numero delle pagine.
Infine una parola sullo scrolling orizzontale. È stato ottenuto semplicemente con l'istruzione: CALL VCHAR (1,1,32,32★24).

Conclusione

Abbiamo visto a grandi linee come funziona il programma, e, nei limiti delle capacità del Texas TI99/4A può essere comodamente applicato per usi concreti, riteniamo. Alcune precisazioni però prima che sia troppo tardi.
Il programma occupa tutta la memoria disponibile del computer, dopo aver lasciato esattamente lo spazio sufficente ad immagazzinare il testo di 10 pagine alla volta. Questo non è ovviamente modificabile, se non in piccoli dettagli, ma con il rischio di andare in **Memory full in...** e perciò perdere tutte e 10 le pagine scritte.

Variabili usate

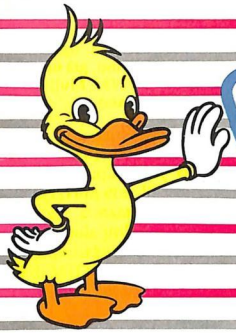
V,H Coordinate di ogni punto della figura iniziale.
COL Variabile corrispondente al Set di colori usati.
R,RIT,LETTURA,UT,RR,CA Variabili di ritorno delle istruzioni CALL KEY usate.
N Numero massimo di pagine da scrivere.
PP Prima pagina da cui partire nella scrittura dei testi.
C Contatore delle pagine usate nel ciclo FOR-NEXT.
I Contatore delle righe di ogni pagina.
RI Prima pagina dalla quale partire nella scrittura (quando si sta aggiornando le pagine già scritte).
A$ (I,C) = Matrice di 10 x 23 punti contenenti le parole.
RED Codice ASCII dei tasti funzione utilizzati nell'operazione di scrittura.
READING, MODYFY Variabili fittizie che discriminano il percorso del programma.
Q$ Chiave per aprire il contenuto del file immagazzinato su cassetta.
Q Numero del file. ■

IL PRIMO SETTIMANALE DI SOFTWARE SU CARTA

L. 1.000

PER IL TUO PERSONAL COMPUTER

Una pubblicazione della J.*soft* editrice



PAPER soft

sped. in abb. post. gr. II/70

apple

TI-99/4A

sinclair Spectrum

C64

VIC 20

In edicola ogni venerdì !

Editrice J.soft via Rosellini, 12 - 20124 Milano - tel. 02/6888228 - 683797 - 6880841 - 6880842 - 6880843

Facile

CALCOLARE...

È facile con MULTIPLAN.
Questo programma per la gestione del "foglio elettronico" trasforma il vostro personal computer in un prestigioso calcolatore che utilizzerete senza problemi. Istruzioni, comandi e ampia documentazione.
Incolonnamento variabile.
Indirizzamento relativo o assoluto. Tutto facilita il lavoro. Completo di guida molto chiara che potete far apparire, a richiesta, sullo schermo.

MODIFICARE...

Desiderate cambiare dei parametri? Multiplan ricalcola automaticamente tutto ciò che ne deriva. La medesima cosa su più fogli di calcolo che potrete legare tra loro a volontà. Integrando funzioni logiche, funzioni statistiche Multiplan si rivela il "foglio elettronico" più potente sul mercato.

DECIDERE...

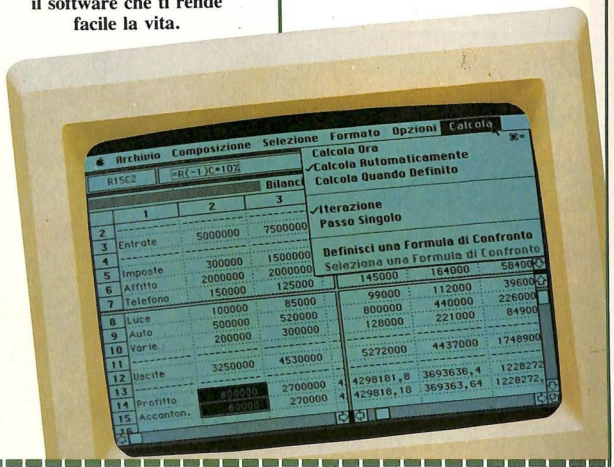
Con Multiplan avrete sotto gli occhi tutte le cifre per prendere decisioni oggettive. Eletto programma dell'anno, già uno dei best-sellers mondiali del 1983, Multiplan, è ora disponibile in Italiano.

con Multiplan

PER APPLE MACINTOSH IN EDIZIONE ITALIANA

MICROSOFT®

il software che ti rende facile la vita.



DISTRIBUITO IN ITALIA
ESCLUSIVAMENTE DA

J.soft EDITRICE

20124 Milano - Via Rosellini, 12
Tel. (02) 6888228-683797-6880841/2/3

Potete acquistare Multiplan - edizione italiana - presso i migliori rivenditori Apple o riceverlo direttamente da J.*soft*, compilando ed inviando il coupon sotto riportato.

Inviare a J.*soft*, - Via Rosellini, 12 - 20124 Milano

Ordino n° ..... Multiplan in italiano per Apple Macintosh - cod. DMOPM02 al prezzo di L. 638.000 cad. (IVA e spese di spedizione incluse).

Scelgo la seguente forma di pagamento:
☐ pagherò in contrassegno al postino
☐ assegno allegato di L. .....

Nome ..... Cognome .....

Via .....

C.A.P. ..... Città ..... Prov. .....

Desidero fattura – n° Partita IVA .....

Data ..... Firma .....

PERSONAL SOFTWARE

Trasformate il vostro Spectrum in un registratore digitale

La proposta SIEL di questo mese è il programma Live Sequencer, realizzato per lo Spectrum con 48 Kbyte di memoria, che consente di registrare in tempo reale un'esecuzione musicale per sintetizzatore o tastiera provvista di MIDI. Questo è possibile purchè si disponga, oltre che dello strumento e del micro di produzione Sinclair, del necessario MIDI Computer Interface SIEL e di un registratore a cassette con cui caricare Live Sequencer (distribuito su cassetta al prezzo di 49.560 lire, IVA inclusa).

Pedali e computer

La riesecuzione della sequenza registrata può essere avviata o interrotta utilizzando un comando a pedale, connesso al SIEL MIDI Computer Interface, lasciando in tal modo libero il musicista di occuparsi d'altro fino al momento opportuno. La capacità di ripetere indefinitamente delle sezioni musicali (ritornelli) esalta gli amanti di musiche ossessive o i perfezionisti in vena di variazioni su basi armoniche ricorrenti. Al Midi Computer Interface SIEL si possono collegare diverse apparecchiature sonore, ma lo strumento da cui si può trarre il migliore profitto è l'Opera 6 della SIEL, visto che Live Sequencer consente di registrare e riascoltare brani polifonici con un totale di sei voci. Ciò non toglie che possa essere convincente anche l'uso della tastiera SIEL MK900 o di altri tipi purchè provvisti di interfaccia MIDI. L'elemento cruciale di questa ibrida catena, vero anello di congiunzione tra applicazione informatica e musica elettronica, è il SIEL MIDI Computer Interface, che consente i necessari collegamenti tra computer e strumenti.

Hardware e operazioni

Il package è stato sviluppato per lo Spectrum Sinclair nella versione a 48 Kbyte di memoria: alla creatura di sir Clive vanno connessi registratore a cassette e televisore, come nelle ordinarie applicazioni d'altro tipo. Una progettazione del software particolarmente curata, con ispirazione dichiaratamente vicina alla filosofia user-friendly, porta a comandi semplificati al punto che spesso è sufficiente premere un solo tasto per attivare una certa funzione. Il menu iniziale permette di scegliere tra le varie funzioni previste dal programma. E' sufficiente rispondere con un numero per attivare ciò che si desidera: Play, ad esempio, corrisponde ad uno e consente di trasferire nella memoria dello Spectrum una sequenza eseguita su una tastiera provvista di MIDI. Il computer resta in questo stato finchè non viene premuto il tasto Break, indice della conclusione della sequenza stessa. In ogni momento sullo schermo è presente un numero che segnala la percentuale di memoria ancora disponibile. Un'occhiata a questo valore consentirà di non rimanere a metà di una sequenza quando si vogliono memorizzare brani molto lunghi, che possono così essere spezzati in più parti separate. Completato il trasferimento, il programma ripropone il menu iniziale, da cui si può scegliere la funzione di riascolto dell'ultima sequenza memorizzata (Playback), corrispondente al due.

Registrazione e caricamento di sequenze

Dopo avere scelto Play è possibile trasferire su cassetta la sequenza che si trova nella memoria centrale dello Spectrum; le modalità di registrazione sono le stesse di un qualsiasi programma. Una volta completata la fase di Save on tape occorre premere il tasto Enter per ritornare al menu. Allo scopo di recuperare una sequenza registrata in precedenza su cassetta e di porla in memoria per poterla riascoltare o ripetere è necessario scegliere tre, Load from tape. Come nel caso del salvataggio, è molto semplice trasferire una sequenza da nastro a memoria: basta compiere le stesse azioni che vi hanno permesso di far partire Live Sequencer, ovviamente dopo aver montato e posizonata la cassetta contenente il brano richiesto. Terminata questa serie di azioni occorre battere Enter per tornare al menu iniziale, dove restano ancora alcune cose da scoprire. In associazione con la funzione di Playback è possibile, premendo i tasti 5 e 8, modificare la velocità d'esecuzione di una sequenza. Premendo 5 la si rallenta, mentre l'8 la fa aumentare; per riportare il brano alla velocità originale occorre tornare all'inizio e battere 5 Enter (Correct time).

Ma è sempre la solita musica...

La funzione di Refrain abilita la ripetizione della sequenza presente in memoria all'infinito: dopo aver premuto 6, occorre scegliere 1 (Yes) e passare poi in modalità Playback, per riascoltare indefinitamente un ritornello. Per chiudere l'ossessionante litania occorre premere Break, e si torna ancora una volta all'inizio. Non dimenticate che si può collegare un pedale al SIEL Midi Computer Interface per avere le mani più libere... SIEL S.p.A. - Zona Industriale - 63030 Acquaviva Picena (AP) - Tel. 0735-607044 ■

PERSONAL SOFTWARE

Testi personalizzati per Sharp MZ 700

Un programma per la stampa di circolari

di *Gatti Giuseppe*

Questo programma permette, a chi possiede una stampante, di ottenere testi personalizzati, circolari a più indirizzi, elenchi alfabetici, ovviamente con memorizzazione e richiamo dati da nastro.
Successivamente, con qualche piccola modifica, potrà essere utilizzato anche con il plotter incorporato nell'MZ 731.
Ora passiamo con ordine al dettaglio del programma: il primo menu visualizzato, non appena dato il RUN è:
A) Carica dati;
B) Modifica/stampa;
C) Record;
D) Indirizzi.

Le opzioni A) e C)

Servono per richiamare o memorizzare i dati da o su cassetta, basterà dare il numero del file (importante ricordarsi il nome esatto del file da caricare per evitare inutili perdite di tempo) e seguire le istruzioni che via via compariranno sullo schermo.
Occorre seguire alcune avvertenze:

- scrivere sulla cassetta il nome del file memorizzato per richiamarlo esattamente;

Listato 1. *Il programma per la stampa di testi e circolari.*

```
10 REM*****************************************
20 REM*****************************************
30 REM      DATA-TESTI-CIRCOLARI         ****
40 REM      BY     GATTI GIUSEPPE        ****
50 REM                                   ****
60 REM           RIVANAZZANO (PV)        ****
70 REM*****************************************
80 REM*****************************************
90 DIMARI$(10):DIMGEO$(10):DIMALGE$(10):DIMSTI(10):DIMINDI$(100,5):DIMSI(100),A$
(21),A2(100),A3(100),NN$(100,5)
100 REM****************************************
110 REM****************************************
120 REM
130 REM   MENU INIZIALE
140 REM
150 REM****************************************
160 REM****************************************
170 CONSOLE0,25,0,40
180 PRINT"E"
190 CURSOR10,5:PRINT" [A]  CARICA DATI"
200 CURSOR10,8:PRINT" [B]  MODIFICA/STAMPA"
210 CURSOR10,11:PRINT" [C]  RECORD  "
220 CURSOR10,14:PRINT" [D] INDIRIZZI"
230 GET%$
240 IF%$=""THEN230
250 IF%$="A"THENGOSUB2920
260 IF%$="B"THEN300
270 IF%$="C"THENGOSUB3320
280 IF%$="D"THEN810
290 GOTO180
300 CONSOLE0,25,0,40:PRINT"E":FORI=1TO10:STI(I)=0:NEXTI:FORI=1TO100:SI(I)=0:NEXT
I
310 CURSOR10,5:PRINT" [1]  CORR. CON INDIRIZZI"
320 CURSOR10,10:PRINT" [2]  CORR. AMICI"
330 CURSOR10,15:PRINT" [3]  CORR. VARIA"
```

Testi personalizzati per Sharp MZ 700

```
340 CURSOR30,20:INPUT"SCELTA  ";Y
350 IFY>3THEN300
360 REM**********************************
370 REM
380 REM  VISUALIZZAZIONE/MODIFICA/STAMPA TESTI CON INDIRIZZI
390 REM
400 REM**********************************
410 PRINT"▣"
420 FORI=1TO4:CURSOR17,I:PRINT"▌ ▌":NEXTI:CURSOR0,5:PRINT"▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
430 CURSOR2,1:PRINT"[M] MODIFICA":CURSOR2,3:PRINT"[S] STAMPA"
440 CONSOLE6,17,0,40
450 ONYGOTO460,1840,2400
460 FORI=1TO10:CURSOR21,1:PRINT"CORR.AFFITTI":CURSOR20,3:PRINTI" TESTO":CURSOR0,
8:PRINT:PRINTARI$(I):PRINT:PRINT:PRINT"PREMI  UN TASTO PER CONTINUARE"
470 GETK$
480 IFK$=""THEN470
490 IFK$="M"THENGOSUB590
500 IFK$="S"THENSTI(I)=1
510 PRINT"▣"
520 NEXTI
530 PRINT"▣▣▣▣▣▣▣▣▣▣ALTRE MODIFICHE  [S/N]-[P]=STAMPA"
540 GETK$
550 IFK$="S"THEN300
560 IFK$="N"THEN170
570 IFK$="P"THENGOSUB680
580 GOTO540
590 PRINT"▣"
600 PRINTI"  TESTO DA MODIFICARE":PRINTARI$(I)
610 INPUT"▣▣▣";ARI$(I):PRINT"▣";ARI$(I)
620 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣VA BENE   [S/N]  "
630 GETZ$
640 IFZ$=""THEN630
650 IFZ$="N"THEN590
660 IFZ$="S"THENRETURN
670 GOTO630
680 CONSOLE0,25,0,40:PRINT"▣▣▣▣▣QUESTO E'IL TESTO CHE STAMPO"
690 FORI=1TO10
700 IFSTI(I)=1THENPRINTARI$(I)
710 NEXTI
720 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣VA BENE  [S/N]"
730 GETK$
740 IFK$=""THEN730
750 IFK$="N"THEN300
760 REM**********************************
770 REM
780 REM  VISUALIZZAZIONE/MODIFICA/STAMPA - INDIRIZZI
790 REM
800 REM**********************************
810 CONSOLE0,25,0,40
820 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣▣ * INDIRIZZI  *"
830 PRINT"▣▣▣▣▣▣▣▣▣▣[1] VISUALIZZARE "
840 PRINT"▣▣▣▣▣▣▣▣▣▣[2] MODIFICARE  "
```

Testi personalizzati per Sharp MZ 700

- se si memorizzano dei testi all'inizio di una cassetta ricordarsi sempre di avvolgere a mano la parte bianca di coda perché l'inizio della registrazione avviene immediatamente dopo aver schiacciato il tasto Record + un tasto qualsiasi del computer;
- non dare ai file nomi complicati o troppo lunghi (max. 10 caratteri);
- sullo schermo, in fase di registrazione o di richiamo, sono visualizzati i testi e gli indirizzi letti o registrati.

Opzione B) Modifica e Stampa

Con l'opzione B) si possono avere:
1) testi con indirizzi;
2) e 3) testi semplici con copie.

Nel primo caso la maschera che comparirà sul video permetterà di visualizzare - modificare - stampare le 10 righe che compongono il testo. Le righe sono in pratica delle stringhe di 250 caratteri che devono sempre iniziare con una lettera del testo o con un simbolo (esempio "—") per non perdere gli spazi bianchi prima del testo vero e proprio in caso di particolare incolonnamento.

Ammettiamo, ad esempio, di non aver ancora in memoria alcun testo.

Seguito listato 1.

```
850 PRINT"■■■■■■■■■■[3] CONTINUARE "
860 PRINT"■■■■■■■■■■[4] MENU "
870 PRINT"■■■■■■■■■■[5] ORDINARE"
880 GETW$
890 IFW$=""THEN880
900 IFW$="1"THEN960
910 IFW$="2"THEN1330
920 IFW$="3"THEN1570
930 IFW$="4"THEN180
940 IFW$="5"THEN3700
950 GOTO880
960 REM VISUALIZZAZIONE INDIRIZZI
970 PRINT"■■■■■■■■■■■■■[S]  STAMPANTE"
980 PRINT"■■■■■■■■■■■■■[M]  MONITOR"
990 GETS$:IFS$=""THEN990
1000 IFS$="M"THEN1200
1010 IFS$="S"THEN1080
1020 GOTO990
1030 REM************************************
1040 REM
1050 REM  STAMPA TABULATO ISCRITTI
1060 REM
1070 REM************************************
1080 PRINT/P"■▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
1090 PRINT/P"          XXXXXXXXXXXXXXXXXXXXXXX          ":PRINT/P
1100 PRINT/P"          XXXXXXXXXXXXXXXXXXXXXXX            ":PRINT/P"            ---------
----------------  "
1110 PRINT/P"■▬▬▬  ELENCO ISCRITTI AL    CLUB    ▬▬▬":PRINT/P:PRINT/P:PRINT/P
1120 I1=1
1130 FORI=1TO100
1140 IF(INDI$(I,2)="")+(LEFT$(INDI$(I,1),3)<>"EGR")THEN1190
1150 PRINT/P"■";I1;" \";INDI$(I,2);"  ";INDI$(I,3);"  ";INDI$(I,4)
1160 PRINT/P
1170 IF(I1=25)+(I1=50)+(I1=75)THENPRINT/P"■"
1180 I1=I1+1
1190 NEXTI:GOTO810
1200 PRINT"■■■■■■■■■■■■[CR] PER CONTINUARE":PRINT"■■■■■■■■■■■[C ] PER CACELLARE"
:PRINT"■■■■■■■■■■■[S ] PER STAMPARE ":PRINT"■■■■■■■■■■■[U ] PER USCIRE "
1210 CONSOLE10,15,0,40
1220 FORI=1TO100:PRINT"■";I;" INDIRIZZO":FORL=1TO4
1230 PRINT"■■■■■■■■■"INDI$(I,L)
1240 NEXTL
1250 GETW$
1260 IFW$=""THEN1250
1270 IFW$="C"THENFORY=1TO4:IND$(I,Y)="":NEXTY
```

Testi personalizzati per Sharp MZ 700

La prima riga sarà pertanto una stringa vuota.
Premere M; comparirà la scritta "Testo da modificare" con il cursore lampeggiante all'inizio della stringa.
Come già detto prima battere la riga (max. 250 caratteri comunque non oltrepassare le 6 righe video).
Terminata la battitura premere CR come un input normale; viene riscritta la stringa così come sarà stampata e la richiesta "Va bene S/N".
In caso affermativo battere S per passare alla riga successiva, se no battere N per ritornare al modo input (con la stringa riscritta) per le modifiche del caso.
Comunque l'uso del programma può chiarire i punti dubbi.
Il tasto CR ha la funzione di ENTER per l'input ed è un tasto valido per avanzamento nella visualizzazione delle righe.

Giunti alla decima riga apparirà l'opzione "Altre modifiche [S/N] [P] stampa".
Se non si hanno più modifiche al testo occorrerà stabilire quali righe dovremo stampare ed allora digitando S ritornerà al menu iniziale.
Battere 1 per riaccedere al pro-

```
1280 IFW$="S"THENSI(I)=1
1290 IFW$="U"THEN810
1300 NEXTI
1310 GOTO820
1320 REM**************************************
1330 REM MODIFICA INDIRIZZI
1340 REM**************************************
1350 FORI=1TO100:PRINT"■■■■■■";I;" INDIRIZZO":FORY=1TO4:PRINT;INDI$(I,Y):NEXTY
1360 PRINT"■ MODIFICO   [S/N]"
1370 GETW$
1380 IFW$=""THEN1370
1390 IFW$="N"THEN1500
1400 IFW$="S"THEN1420
1410 GOTO1370
1420 INPUT"■■TITOLO ONORIF.  ";INDI$(I,1)
1430 INPUT"NOME COGNOME     ";INDI$(I,2)
1440 INPUT"VIA              ";INDI$(I,3)
1450 INPUT"CAP CITTA' PROV.";INDI$(I,4)
1460 PRINT"■■■■■■■■ALTRE MODIFICHE [S/N]"
1470 GETW$
1480 IFW$=""THEN1470
1490 IFW$="N"THEN820
1500 NEXTI
1510 GOTO820
1520 REM**************************************
1530 REM
1540 REM ROUTINE STAMPA CIRCOLARI
1550 REM
1560 REM**************************************
1570 INPUT"■■■■■■■■■■■■QUANTI SPAZI DOPO L'INDIRI. ";SPAZI
1580 INPUT"■■■■■■■■■■■■■■■■■CARATTERE INDRIZZO [G/P]  ";CI$
1590 INPUT"■■■■■■■■■■■■■■■■■CARATTERE TESTO  [G/P]  "   ;CT$
1600 FORL=1TO100
1610 IFSI(L)=0THEN1760
1620 PRINT/P
1630 PRINT/P"■■■■■■■■■■■■■■■■■
```

Seguito listato 1.

```
1710 IFSTI(T)=0THEN1740
1720 IFCT$="G"THEN PRINT#P"■";ARI$(T)
1730 IFCT$="P"THEN PRINT#P"■";ARI$(T)
1740 NEXTT
1750 PRINT#P"■"
1760 NEXTL
1770 PRINT"■■■■■■■■■■■■■■■■■STESSO TESTO   [S/N]"
1780 RETURN
1790 REM**************************************
1800 REM
1810 REM  VISUALIZZAZIONE/MODIFICA/STAMPA  TESTI SENZA INDIRIZZI
1820 REM
1830 REM**************************************
1840 FORI=1TO10:CURSOR21,1:PRINT"CORR.AMICI":CURSOR20,3:PRINTI" TESTO":CURSOR0,8
:PRINT:PRINTGEO$(I):PRINT:PRINT:PRINT"PREMI  UN TASTO PER CONTINUARE"
1850 GETK$
1860 IFK$=""THEN1850
1870 IFK$="M"THENGOSUB1970
1880 IFK$="S"THENSTI(I)=1
1890 PRINT"■"
1900 NEXTI
1910 PRINT"■■■■■■■■■■■ALTRE MODIFICHE  [S/N]-[P]=STAMPA"
1920 GETK$
1930 IFK$="S"THEN300
1940 IFK$="N"THEN170
1950 IFK$="P"THENGOSUB2070
1960 GOTO1920
1970 PRINT"■"
1980 PRINTI"  TESTO DA MODIFICARE":PRINTGEO$(I)
1990 INPUT"■■■";GEO$(I):PRINT"■";GEO$(I)
2000 PRINT"■■■■■■■■■■■■VA BENE   [S/N]  "
2010 GETZ$
2020 IFZ$=""THEN2010
2030 IFZ$="N"THEN1970
2040 IFZ$="S"THENRETURN
2050 GOTO2010
2060 REM**************************************
2070 REM ROUTINE DI STAMPA
2080 REM**************************************
2090 CONSOLE0,25,0,40::PRINT"■■■■■QUESTO E' IL TESTO CHE STAMPO"
2100 FORI=1TO10
2110 IFSTI(I)=1THENPRINTGEO$(I)
2120 NEXTI
2130 PRINT"■■■■■■■■■■■■■VA BENE  [S/N]"
2140 GETK$
2150 IFK$=""THEN2140
2160 IFK$="N"THEN300
2170 INPUT"■■■■■■■■■■■■■■■QUANTE COPIE ";COP
2180 INPUT"■■■■■■■■■■■■■■QUANTI SPAZI TRA I TESTI   ";SPAZI
2190 INPUT"■■■■■■■■■■■■■■CARATTERE TESTO  [G/P]  ";CT$
2200 FORL=1TOCOP
2210 PRINT#P"■"
2220 FORI=1TO10
2230 IFSTI(I)<>1THEN2260
2240 IFCT$="G"THENPRINT#P"■"GEO$(I)
2250 IFCT$="P"THENPRINT#P"■"GEO$(I)
2260 IFSPAZI<1THEN 2280
2270 FORS=1TOSPAZI:PRINT#P:NEXTS
```

Testi personalizzati per Sharp MZ 700

gramma con indirizzi e, alla visualizzazione delle righe, premere S per definire quelle da stampare per proseguire nel programma.
Tutte le volte che si passa da menu testi, le variabili di stampa vengono azzerate, pertanto l'ultimo passaggio dovrà avvenire con il tasto S.
Giunti alla decima riga utilizzando il tasto S per l'avanzamento, accedere alla routine di stampa con il tasto P.
Sul video comparirà il testo che sarà stampato, ovviamente con l'incolonnamento per le 40 colonne video.
Battere S per continuare se il testo va bene.
Battere N per ritornare al menu testi se il testo discorda o è nullo.
Nell'ultimo caso vale quanto già detto e pertanto, dopo il menu lo scorrimento delle righe dovrà avvenire con il tasto S per quelle da stampare, oppure un tasto qualsiasi per quelle da escludere dalla stampa.
Il passaggio successivo è la routine indirizzi a cui si può accedere dopo la visualizzazione del testo oppure direttamente dal menu iniziale con l'opzione D indirizzi.

```
2280 NEXTI
2290 NEXTL
2300 PRINT"■■■■■■■■■■■■■■■■■■■■STESSO ARGOMENTO [S/N] "
2310 REM**************************************
2320 REMFINE ROUTINE DI STAMPA
2330 REM**************************************
2340 RETURN
2350 REM**************************************
2360 REM
2370 REM VISUALIZZAZIONE/MODIFICA/STAMPA  TESTI SENZA INDIRIZZI
2380 REM
2390 REM**************************************
2400 FORI=1TO10:CURSOR21,1:PRINT"CORR.VARIA":CURSOR20,3:PRINTI" TESTO":CURSOR0,8
:PRINT:PRINTALGE$(I):PRINT:PRINT:PRINT"PREMI  UN TASTO PER CONTINUARE"
2410 GETK$
2420 IFK$=""THEN2410
2430 IFK$="M"THENGOSUB2530
2440 IFK$="S"THENSTI(I)=1
2450 PRINT"■"
2460 NEXTI
2470 PRINT"■■■■■■■■■■■■■■■■ALTRE MODIFICHE  [S/N]-[P]=STAMPA"
2480 GETK$
2490 IFK$="S"THEN300
2500 IFK$="N"THEN170
2510 IFK$="P"THENGOSUB2630
2520 GOTO2480
2530 PRINT"■"
2540 PRINTI"  TESTO DA MOFIFICARE":PRINTALGE$(I)
2550 INPUT"■■■";ALGE$(I):PRINT"■";ALGE$(I)
2560 PRINT"■■■■■■■■■■■■VA BENE   [S/N]  "
2570 GETZ$
2580 IFZ$=""THEN2570
2590 IFZ$="N"THEN2530
2600 IFZ$="S"THENRETURN
2610 GOTO2570
2620 REM**************************************
2630 REM ROUTINE DI STAMPA
2640 REM**************************************
2650 CONSOLE0,25,0,40:PRINT"■■■■QUESTO E' IL TESTO CHE STAMPO"
2660 FORI=1TO10
2670 IFSTI(I)=1THENPRINTALGE$(I)
2680 NEXTI
2690 PRINT"■■■■■■■■■■VA BENE  [S/N]"
2700 GETK$
2710 IFK$=""THEN2700
2720 IFK$="N"THEN300
2730 INPUT"■■■■■■■■■■■■■■■■■■QUANTE COPIE  ";COP
2740 INPUT"■■■■■■■■■■■■■QUANTI SPAZI TRA I TESTI   ";SPAZI
```

Seguito listato 1.

```
2750 FORL=1 TOCOP
2760 PRINT#P"■"
2770 FORI=1TO10
2780 IFSTI(I)<>1THEN2820
2790 IFCT$="G"THENPRINT#P"■";ALGE$(I)
2800 IFCT$="P"THENPRINT#P"■";ALGE$(I)
2810 FORV=0TOSPAZI:PRINT#P:NEXTV
2820 NEXTI
2830 NEXTL
2840 PRINT"■■■■■■■■■■■■■■■■■■STESSO ARGOMENTO  [S/N] "
2850 REMFINE ROUTINE DI STAMPA
2860 RETURN
2870 REM**************************************
2880 REM
2890 REM  ROUTINE DI PRELIEVO DATI DA CASSETTA
2900 REM
2910 REM**************************************
2920 PRINT"■■■■■■■■■■■■■■■■RIAVVOLGI LA CASSETTA LATO  B":PRINT"■■■■■■■■■■■■■■■■■■F
ATTO   [S/N]"
2930 GETW$
2940 IFW$=""THEN2930
2950 IFW$="S"THEN2980
2960 IFW$="N"THENRETURN
2970 GOTO2930
2980 INPUT"■■■■■■■■■■■■■■■■■■■■■■■■NOME RECORD  ";REC$
2990 PRINT"■■■■■■■■■■■■■■■■PREMI PLAY E SCHIACCIA UN TASTO"
3000 GETW$
3010 IFW$=""THEN3000
3020 ROPENREC$
3030 FORI=1TO10
3040 INPUT#TARI$(I)
3050 PRINT"■■■■■HO LETTO QUESTO TESTO"
3060 PRINTARI$(I)
3070 INPUT#TGEO$(I)
3080 PRINT"■■■■HO LETTO QUESTO TESTO"
3090 PRINTGEO$(I)
3100 INPUT#TALGE$(I)
3110 PRINT"■■■■HO LETTO QUESTO TESTO"
3120 PRINTALGE$(I)
3130 NEXTI
3140 CLOSE
3150 ROPEN"INDIRIZZI"
3160 FORI=1TO100
3170 FORV=1TO5
3180 INPUT#TINDI$(I,V)
3190 NEXTV:PRINTI;"  INRIRIZZO":
3200 FORV=1TO5: PRINTINDI$(I,V):NEXTV
3210 NEXTI:CLOSE
3220 PRINT"■■■■■■■■■■■■■■■■■HO ULTIMATO LA LETTURA":PRINT"■■■■■■■■PREMI  STOP  "
3230 PRINT"■■■■■PREMI UN TASTO PER CONTINUARE"
3240 GETW$
3250 IFW$=""THEN3240
3260 RETURN
3270 REM**************************************
3280 REM
3290 REM  ROUTINE DI RECORD DATI SU CASSETTA
3300 REM
3310 REM**************************************
```

```
3320 PRINT"■■■■■■■■■■■■■■■■RIAVVOLGI LA CASSETTA LATO  B":PRINT"■■■■■■■■■■■■■■■F
ATTO   [S/N]"
3330 GETW$
3340 IFW$=""THEN3330
3350 IFW$="S"THEN3380
3360 IFW$="N"THENRETURN
3370 GOTO3330
3380 INPUT"■■■■■■■■■■■■■■■■■■■■NOME RECORD ";REC$
3390 PRINT"■■■■■■■■■■■■■■■■PREMI RECORD E SCHIACCIA UN TASTO"
3400 GETW$
3410 IFW$=""THEN3400
3420 WOPENREC$
3430 FORI=1TO10
3440 PRINT"■■■■REGISTRO QUESTO TESTO"
3450 PRINTARI$(I)
3460 PRINT#TARI$(I)
3470 PRINT"■■■■REGISTRO QUESTO TESTO"
3480 PRINTGEO$(I)
3490 PRINT#TGEO$(I)
3500 PRINT"■■■■REGISTRO QUESTO TESTO"
3510 PRINTALGE$(I)
3520 PRINT#TALGE$(I)
3530 NEXTI:CLOSE
3540 WOPEN"INDIRIZZI"
3550 FORI=1TO100:PRINTI;"   INDIRIZZO"
3560 FORY=1TO5
3570 PRINT#TINDI$(I,Y):PRINTINDI$(I,Y)
3580 NEXTY
3590 NEXTI
3600 CLOSE
3610 PRINT"■■■■■■■■■■■■■■HO FINITO PREMI STOP POI UN TASTO"
3620 GETW$
3630 IFW$=""THEN3620
3640 RETURN
3650 REM**************************************
3660 REM
3670 REM  ROUTINE DI ORDINAMENTO ALFABETICO ELENCO INDIRIZZI
3680 REM
3690 REM**************************************
3700 PRINT"■■■■■■■■■■■■■■STO ORDINANDO L'ARCHIVIO"
3710 PRINT"■■■■■■■■■■■■■■ATTENDI PREGO...."
3720 A$(1)="A":A$(2)="B":A$(3)="C":A$(4)="D":A$(5)="E":A$(6)="F":A$(7)="G":A$(8)
="H"
3730 A$(9)="I":A$(10)="L":A$(11)="M":A$(12)="N":A$(13)="O":A$(14)="P":A$(15)="Q"
:A$(16)="R"
3740 A$(17)="S":A$(18)="T":A$(19)="U":A$(20)="V":A$(21)="Z"
3750 K1=0:FORI=1TO100:IFINDI$(I,2)<>""THENK1=K1+1
```

1) Visualizzazione indirizzi:
S) stampante - M) monitor.
L'opzione S permette la stampa di un tabulato personalizzato (personalizzazione di cui tratteremo più avanti) di tutti gli indirizzi che hanno nel titolo onorifico le prime 3 lettere ="EGR".
Questa scelta è stata inserita per elencare nel tabulato solo alcuni nominativi, mentre le lettere potrebbero essere inviate anche ad altre persone non a titolo ricorrente, ma comunque da inserire nella rosa degli indirizzi.
L'opzione M visualizzerà uno ad uno tutti gli indirizzi sul video con le seguenti possibilità (riportate in alto sullo schermo):
CR per continuare;
C per cancellare;
S per stampare;
U per uscire.
L'opzione S ha la stessa funzione delle righe di testo e determina gli indirizzi da stampare.
L'opzione U serve per ritornare al menu indirizzi.
2) Modificare: serve per inserire o modificare un indirizzo.
Sul video sarà visualizzato l'indirizzo (o una stringa vuota se non vi sono elementi) con la richiesta "Modifico S/N".
Digitando S escono ad uno ad uno gli input dei campi dell'indirizzo con richiesta finale di conferma.
Per il motivo già detto alcuni nominativi avranno un titolo onorifico che inizia con "EGR", gli altri un altro titolo (esempio Ill.mo).
Digitando N si passa alla visualizzazione dell'indirizzo successivo.
Per uscire dalla routine occorre far passare tutti i 100 indirizzi con il tasto N che lascerà inalterato l'archivio.
3) Continuare: ultimata la scelta degli indirizzi da stampare con questa opzione si passa alla routine di stampa.
4) Menu: ritorno al menu iniziale del programma.
5) Ordinare: riordina l'archivio indi-

Seguito listato 1.

```
3760 FORK=1TO5
3770 NN$(I,K)=""
3780 NEXTK
3790 NEXTI
3800 FORI=1TOK1
3810 FORL=1TO21
3820 IFINDI$(I,2)=""THEN3880
3830 IFINDI$(I,5)<>""THEN3880
3840 IFA$(L)=LEFT$(INDI$(I,2),1)THENA2(I)=25*L
3850 IFA$(L)=MID$(INDI$(I,2),2,1)THENA3(I)=L
3860 NEXTL
3870 INDI$(I,5)=STR$(A2(I)+A3(I))
3880 NEXTI
3890 XV=1
3900 FORI=1TO550
3910 FORL=1TOK1
3920 IFINDI$(L,2)=""THEN3950
3930 IFVAL(INDI$(L,5))=ITHENFORK=1TO5:NN$(XV,K)=INDI$(L,K):INDI$(L,K)="":NEXTK:X
V=XV+1
3940 PRINT"■■";L
3950 NEXTL
3960 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";I
3970 NEXTI
3980 FORI=1TO100
3990 FORL=1TO5
4000 INDI$(I,L)=NN$(I,L)
4010 NEXTL
4020 NEXTI
4030 GOTO810
```

COMMENTO LINEE

10-90	Dimensionamento matrici e variabili.
100-290	Menu iniziale e rimandi alle linee del programma.
300-450	Opzione testi e rimando linee.
460-750	Stesura - correzione - stampa testo con indirizzi.
810-1020	Stesura - correzione - visualizzazione - stampa indirizzi.
1030-1190	Stampa tabulato iscritti - Attenzione - stampa solo quelli che iniziano con Egr. sig.
1200-1310	Visualizzazione indirizzi su monitor con opzione per stampa - cancellazione.
1320-1510	Routine modifica indirizzi.
1520-1780	Stampa lettere con gli indirizzi stabiliti.
1790-2050	Stesura - modifica - stampa testi senza indirizzi.
2060-2340	Stampa testo con opzione del numero delle copie.
2350-2610	Stesura - modifica - stampa testo senza indirizzi.
2620-2860	Stampa testo con opzione del numero delle copie.
2870-3260	Lettura archivio file da cassetta con visualizzazione dati letti.
3270-3640	Registrazione archivio testi e indirizzi su cassetta.
3650-4030	Ordinamento alfabetico dell'archivio indirizzi.

ELENCO VARIABILI

ARI$(10)	Testi con indirizzi GEO$(10) = testi semplici.
ALGE$(10)	Testi semplici.
STI(10)	1 = testo da stampare; 0 testo da non stampare.
INDI$(100,5)	Indirizzi con 5 campi, il quinto serve per il numero di ordinamento alfabetico.
MSI(100)	1 = indirizzo da stampare; 0 = indirizzo da non stampare.

rizzi con ordine alfabetico sulle prime due lettere del cognome.

Routine di stampa

All'inizio della routine di stampa vi sono 3 input per determinare:
- gli spazi tra indirizzo e testo;
- il carattere degli indirizzi P piccolo G grosso;
- il carattere del testo P piccolo G grosso.

Prima di dare il CR all'input carattere testo assicurarsi che la stampante sia accesa e posizionata all'inizio del foglio.
Dopo il CR la stampante inizierà la stampa del testo modificando, ad ogni foglio, l'indirizzo.
Le copie stampate saranno esattamente uguali agli indirizzi scelti.

Personalizzazione testi

È possibile personalizzare i fogli da spedire o il tabulato elenco iscritti con le indicazioni del club - dell'associazione o altro.
Per fare ciò occorre modificare il programma alle righe 1090-1100 per il tabulato iscritti; alle righe 1640 - 1650 per le lettere, sostituendo alle XXXXXXXX il nome del club - la città ecc. ■

J.soft

CERCA:

ambosessi di qualsiasi età, residenti in Italia o all'estero

REQUISITI RICHIESTI:

forte interesse per gli home computer Commodore o Sinclair disponibilità tempo libero per appassionante lettura di "superivista" dedicata

OFFRE:

abbonamento a 11 numeri di SuperSinc o SuperVic al **prezzo speciale** di
L. 66.000
compresa cassetta
oppure
L. 30.000
per la sola rivista

inviando il coupon pubblicato a fondo pagina otterrete pronta soddisfazione alla vostra richiesta.

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure ritagliate il tagliando abbonamenti pubblicato in questa pagina e speditelo allegando un assegno intestato a: *J. soft - Via Rosellini 12 - 20124 Milano.*

Tagliando abbonamento a SUPERVIC e SUPERSINC da inviare in busta chiusa a: *J.soft* Via Rosellini 12 - 20124 Milano

- ☐ Abbonamento a 11 numeri di SUPERVIC al prezzo speciale di L. 30.000
- ☐ Abbonamento a 11 numeri di SUPERVIC + cassetta con tutti i programmi pubblicati al prezzo speciale di L. 66.000
- ☐ Abbonamento a 11 numeri di SUPERSINC al prezzo speciale di L. 30.000
- ☐ Abbonamento a 11 numeri di SUPERSINC + cassetta con tutti i programmi pubblicati al prezzo speciale di L. 66.000

cognome _____ nome _____

via _____ città _____

cap. _____ provincia _____ data _____

firma _____

PERSONAL SOFTWARE

Alta risoluzione

Un valido aiuto per la grafica del C 64

di *Antonio Bartalesi*

Il Commodore 64 possiede delle discrete capacità grafiche, per esempio in alta risoluzione (bit map mode) permette di controllare lo schermo con una risoluzione di 320 x 200 punti, e fornisce anche un certo controllo dei colori.

Purtroppo la grafica in alta risoluzione non è stata implementata nel sistema operativo, e ciò costringe chi vuole usarla e non dispone di un apposito package in linguaggio macchina, a scrivere lunghi, e lenti, programmi in BASIC pieni di istruzioni PEEK e POKE.
Comunque, ciò che spesso scoraggia maggiormente l'uso dell'alta risoluzione, è la necessità di eseguire frequentemente complesse operazioni di PEEK e POKE per passare dalla pagina in alta risoluzione a quella testo, senza peraltro poterle mai vedere entrambe contemporaneamente.

La routine descritta in questo articolo fornisce una soluzione molto semplice a questo problema: la pagina in alta risoluzione può in qualsiasi momento, anche durante l'esecuzione di un programma, essere calata su e giù come un sipario su quella testo, e fermata a qualsiasi altezza.

Ciò permette, per esempio, di creare un disegno in alta risoluzione continuando ad usare le ultime linee di schermo in basso per esaminare il programma o per eseguire delle istruzioni di input.
La routine non elimina naturalmente le difficoltà intrinseche all'uso dell'alta risoluzione, vuole piuttosto

Listato 1. *Il programma in Assembly.*

```
10000 ---------------------------------
10010 HIRES SYSTEM.
10020 ;
10030 SI DEFINISCE LA PARTE SUPERIORE
10040 DELLO SCHERMO IN ALTA RISOLUZIONE
10050 E LA PARTE INFERIORE IN MODO
10060 TESTO.
10070 ;
10080 I TASTI CTRL-F5 E CTRL-F7
10090 SPOSTANO IL CONFINE.
10100 CTRL-F1 SPENGE L'ALTA RISOLUZIONE
10110 E CTRL-F3 LA RIACCENDE.
10120 ;
10130 CON L'ALTA RISOLUZIONE ACCESA, LA
10140 SORGENTE PER L'INTERRUPT E' IL
10150 RASTER DEL VIC A 50HZ, ANZICHE'
10160 IL TIMER DELLA CIA A 60HZ.
10170 ;
10180 L'ENTRY POINT E' KEYON; PER
10190 SPENGERE, KEYOFF
10192 ;
10193 L'ALTA RISOLUZIONE E' MAPPATA
10194 NELLA RAM DA $6000 A $7FFF.
10195 I COLORI SONO NELL'AREA
10196 DA $5C00 A $5FFF. LA ROUTINE
10197 PARTE DA $9000.
10200 ---------------------------------
10210 DATA"      .ORG $8000
10300 ---------------------------------
10310 DEFINIZIONE COSTANTI ESTERNE.
10320 ;
10330 DATA"VIC   .EQU $D000
10340 DATA"CIA1  .EQU $DC00
10350 DATA"CIA2  .EQU $DD00
10360 DATA"CINV  .EQU $314
10370 DATA"KEYLOG.EQU $28F
10380 DATA"IRQK  .EQU $EA31
10390 DATA"SFDX  .EQU $CB
10400 DATA"LSTX  .EQU $C5
10410 DATA"SHFLAG.EQU $28D
10500 ---------------------------------
10510 PARAMETRI E VARIABILI.
10520 ;
10530 DATA"BANK  .BYT 2,3
10540 DATA"MEMPTS.BYT $78,$10
10560 DATA"RASTER.BYT $C0,0
10580 DATA"DELAY .BUF 1
10590 DATA"FREQ  .BUF 1
10600 ---------------------------------
10610 SCAMBIO ALTA/BASSA RISOLUZIONE.
10620 ;
10630 DATA"SWAP    LDX #0
10640 DATA"        LDA VIC+17
10650 DATA"        AND #$20
10660 DATA"        BEQ A1
10670 DATA"        INX
10680 DATA"A1      ;
10690 X=0 NOW LORES : X=1 NOW HIRES
10700 ;
10710 PONE IL RASTER LATCH PER LA
10720 PROSSIMA INTERRUZIONE E CAMBIA IL
10725 MODO GRAFICO:
10730 DATA"        LDA RASTER,X
10740 DATA"        STA VIC+18
10750 DATA"        LDA VIC+17
10760 DATA"        AND #$7F
```

togliere un ostacolo di tipo psicologico.
Il programma, in linguaggio macchina, è stato realizzato usando un assemblatore scritto in BASIC e, anziché presentare soltanto una sfilza di numeri senza significato in istruzioni DATA, si è preferito riportare direttamente il listato in Assembly commentato (listato 1), anche se la sua sintassi difficilmente sarà accettata da un altro assemblatore.
Per chi possiede un assemblatore, non sarà comunque difficile interpretare e riscrivere le istruzioni. A chi ha un monitor potrà essere utile la tabella dei simboli nel listato 2.

Per dare anche agli altri la possibilità di usare il programma, nel listato 3 è riportato un elenco di istruzioni DATA, i cui codici devono essere inseriti in memoria a partire dalla locazione 32768.
Ogni quattro linee DATA, vi è un codice di controllo uguale alla somma dei codici delle quattro linee stesse.
Il lettore potrà aggiungere un programma caricatore in BASIC.
Il listato in Assembly servirà comunque come documentazione, e come un esempio in seguito agli articoli sulla programmazione del 6502 pubblicati su **Personal Software**.

Uso del programma

La pagina in alta risoluzione è quella da $6000 a $7FFF.
Il programma è allocato a partire da $8000 ed è lungo 291 byte.
Per i colori si usa l'area da $5C00 a $5FFF.
Quindi è opportuno limitare l'area di memoria riservata ai programmi e alle variabili BASIC al di sotto di $5C00, eseguendo il comando:
POKE 52,92: POKE 56,92.
La routine, una volta caricata nelle giuste locazioni di memoria, si attiva con SYS 32950, e si disattiva con SYS 32964.
Quando la routine è attiva, funzio-

```
10780 DATA"        EOR #$20
10790 DATA"        STA VIC+17
10800 ;
10810 PONE I PUNTATORI ALLE AREE DI
10820 MEMORIA:
10830 DATA"        LDA VIC+24
10840 DATA"        AND #$07
10850 DATA"        ORA MEMPTS,X
10860 DATA"        STA VIC+24
10870 ;
10880 SELEZIONA IL BLOCCO DI MEMORIA:
10890 DATA"        LDA CIA2+2
10900 DATA"        ORA #$03
10910 DATA"        STA CIA2+2
10920 DATA"        LDA CIA2
10930 DATA"        AND #$FC
10940 DATA"        ORA BANK,X
10950 DATA"        STA CIA2
10960 ;
10970 DATA"        DEX
10980 X=0 NOW LORES : X=-1 NOW HIRES
10985 ;
10990 DATA"        RTS
11000 --------------------------------
11010 ROUTINE DI INTERRUPT RICHIAMATA
11020 DUE VOLTE OGNI SEMIQUADRO, CHE
11023 PRODUCE LA PARTIZIONE DELLO
11026 SCHERMO.
11030 ;
11033 DATA"IRQHIR;
11036 ;
11040 CONTROLLA CHE L'INTERRUPT VENGA
11050 DAL VIC E LO AZZERA:
11060 DATA"        LDA VIC+25
```

```
11070 DATA"        AND #$01
11080 DATA"        BEQ B1
11090 DATA"        STA VIC+25
11100 ;
11110 PER SINCRONIZZARE IL CAMBIO DI
11120 MODO GRAFICO CON LA FINE DI UNA
11130 RIGA:
11140 DATA"SYNC    LDX #5
11150 DATA"B2      DEX
11160 DATA"        BPL B2
11170 IL VALORE INIZIALE VA DETERMINATO
11180 IN FASE DI MESSA A PUNTO.
11190 ;
11200 CAMBIA MODO GRAFICO E UNA VOLTA
11210 OGNI DUE SALTA ALL'INTERRUPT
11220 NORMALE:
11230 DATA"        JSR SWAP
11240 DATA"        BEQ B3
11250 DATA"B1      JMP IRQK
11260 DATA"B3      PLA
11270 DATA"        TAY
11280 DATA"        PLA
11290 DATA"        TAX
11300 DATA"        PLA
11310 DATA"        RTI
11400 --------------------------------
11410 ACCENSIONE DELLA ROUTINE
11420 PRECEDENTE.
11430 ;
11440 DATA"HIRON   PHP
11445 DATA"        SEI
11450 ;
11460 MODIFICA IL VETTORE DI INTERRUPT:
11470 DATA"        LDA #<IRQHIR
```

Seguito listato 1.

```
11480 DATA"        STA CINV
11490 DATA"        LDA #>IRQHIR
11500 DATA"        STA CINV+1
11510 ;
11520 DISABILITA L'INTERRUPT DELLA CIA:
11530 DATA"        LDA #$01
11540 DATA"        STA CIA1+13
11550 ;
11560 ABILITA L'INTERRUPT DEL VIC:
11570 DATA"        ORA VIC+26
11580 DATA"        STA VIC+26
11590 ;
11600 DATA"        PLP
11610 DATA"        RTS
11700 --------------------------------
11710 SPEGNIMENTO DELLA STESSA.
11720 ;
11730 DATA"HIROFF  PHP
11735 DATA"        SEI
11740 ;
11750 RIPRISTINA IL VETTORE STANDARD:
11760 DATA"        LDA #<IRQK
11770 DATA"        STA CINV
11780 DATA"        LDA #>IRQK
11790 DATA"        STA CINV+1
11800 ;
11810 DISABILITA L'INTERRUPT DEL VIC:
11820 DATA"        LDA VIC+26
11830 DATA"        AND #$FE
11840 DATA"        STA VIC+26
11850 ;
11860 RIABILITA QUELLO DELLA CIA:
11870 DATA"        LDA #$81
11880 DATA"        STA CIA1+13
11890 ;
11900 DATA"        PLP
11910 ;
11920 SI ASSICURA CHE LO SCHERMO SIA
11930 RIMASTO IN MODO TESTO:
11940 DATA"C1      JSR SWAP
11950 DATA"        BNE C1
11960 DATA"        RTS
12000 --------------------------------
12010 SPOSTA LA PARTIZIONE DELLO
12020 SCHERMO DI 8 PIXEL VERSO L'ALTO.
12030 ;
12040 DATA"UP      LDA RASTER
12050 DATA"        SEC
12060 DATA"        SBC #8
12070 DATA"        CMP #$30
12080 DATA"        BCS D1
12090 DATA"        LDA #$30
12100 DATA"D1      STA RASTER
12110 DATA"        RTS
12200 --------------------------------
12210 LA SPOSTA DI 8 PIXEL VERSO IL
12220 BASSO.
12230 ;
12240 DATA"DOWN    LDA RASTER
12250 DATA"        CLC
12260 DATA"        ADC #8
12270 DATA"        BCC D2
12280 DATA"        LDA #$F8
12290 DATA"D2      STA RASTER
12300 DATA"        RTS
12400 --------------------------------
12410 ACCENSIONE DEL SISTEMA PER
12420 L'ESAME DEI TASTI FUNZIONE.
12430 ;
12433 MODIFICA IL VETTORE ALLA ROUTINE
12436 CHE ESAMINA IL TASTO PREMUTO:
12440 DATA"KEYON   PHP
12445 DATA"        SEI
12450 DATA"        LDA #<UDKEY
12460 DATA"        STA KEYLOG
12470 DATA"        LDA #>UDKEY
12480 DATA"        STA KEYLOG+1
12490 DATA"        PLP
12500 DATA"        RTS
12600 --------------------------------
12610 SPEGNIMENTO DEL SISTEMA.
12620 ;
12630 RIPRISTINA IL VETTORE STANDARD:
12640 DATA"KEYOFF  PHP
12645 DATA"        SEI
12650 DATA"        LDA #<$EB48
12660 DATA"        STA KEYLOG
12670 DATA"        LDA #>$EB48
12680 DATA"        STA KEYLOG+1
12690 DATA"        PLP
12700 DATA"        RTS
12800 --------------------------------
12810 ROUTINE PER L'ESAME DEI TASTI
12820 FUNZIONE, RICHIAMATA SOTTO
12830 INERRUPT DALLA ROUTINE DI
12840 SCANSIONE TASTIERA.
12850 ;
12860 DATA"UDKEY ;
12870 ;
12880 CONTROLLA CHE SIA PREMUTO IL
12890 TASTO CTRL:
12900 DATA"        LDA SHFLAG
12910 DATA"        CMP #4
12920 DATA"        BNE E1
12930 ;
12940 CONTROLLA CHE SIA PREMUTO UN
12950 TASTO FUNZIONE:
12960 DATA"        LDA Z,SFDX
12970 DATA"        CMP #3
12980 DATA"        BCC E1
12990 DATA"        CMP #7
13000 DATA"        BCS E1
13010 ;
13020 IN QUESTO CASO, INTERPRETA ED
13030 ESEGUE IL COMANDO
13032 (A CONTIENE I SEGUENTI VALORI:
13033 ;3 PER CTRL-F7, 4 PER CTRL-F1,
13034 ;5 PER CTRL-F3, 6 PER CTRL-F5):
13040 DATA"        JSR EXEC
13050 ;
13060 RIENTRA NELLA ROUTINE DI
13070 SCANSIONE TASTIERA:
13080 DATA"E1      JMP $EB48
13100 --------------------------------
13110 INTERPRETAZIONE DEI TASTI.
13120 ;
13130 IL LORO SIGNIFICATO E':
13150 HIROFF PER CTRL-F1,
13160 HIRON  PER CTRL-F3,
13170 UP     PER CTRL-F5,
13180 DOWN   PER CTRL-F7.
13190 ;
```

Seguito listato 1.

```
13200 DATA"EXEC    TAX
13210 ;
13220 SE IL TASTO NON ERA GIA' PREMUTO,
13230 SALTA LA LOGICA DI REPEAT:
13240 DATA"        CMP Z,LSTX
13250 DATA"        BNE F1
13260 ;
13270 SOLO CTRL-F5 E CTR-F7 SI ESEGUONO
13280 ANCORA SE ERANO GIA' PREMUTI:
13290 DATA"        AND #$02
13300 DATA"        BEQ F2
13310 ;
13320 LOGICA DI REPEAT:
13330 DATA"        DEC DELAY
13340 DATA"        BNE F2
13350 DATA"        INC DELAY
13360 DATA"        DEC FREQ
13370 DATA"        BNE F2
13380 DATA"        LDA #4
13390 DATA"        STA FREQ
13400 ;
13410 ESECUZIONE DI CTRL-5 E CTRL-7:
13420 DATA"F3      TXA
13430 DATA"        ROR A
13440 DATA"        BCC UP
13450 DATA"        BCS DOWN
13460 ;
13470 IL TASTO NON ERA GIA' PREMUTO.
13480 DATA"F1      LDY #16
13490 DATA"        STY DELAY
13493 DATA"        LDY #1
13496 DATA"        STY FREQ
13500 ;
13510 SI ESEGUE IL COMANDO:
13520 DATA"        AND #$02
13530 DATA"        BNE F3
13535 DATA"        TXA
13540 DATA"        ROR A
13550 DATA"        BCC F4
13560 DATA"        JMP HIRON
13565 DATA"F4      JMP HIROFF
13570 ;
13580 DATA"F2      RTS
13600 ---------------------------------
13610 DATA"        .END
13700 ---------------------------------
```

nano i tasti CTRL-F1, CTRL-F3, CTRL-F5 e CTRL-F7, col seguente significato: CTRL-F3 abilita la pagina in alta risoluzione ad essere visualizzata, CTRL-F1 la disabilita, CTRL-F5 e CTRL-F7 muovono il sipario rispettivamente verso l'alto e verso il basso.
Queste combinazioni di tasti non dovrebbero interferire con nessun programma scritto in BASIC standard, in quanto ad esse il sistema operativo non assegna codici ASCII.

Descrizione di massima del programma

Il programma è formato da due parti, entrambe eseguite sotto interrupt: una presiede al controllo del VIC (Video Interface Chip), e l'altra all'esame dei tasti.
La routine che controlla il VIC è Irqhir, e viene accesa e spenta rispettivamente da Hiron e Hiroff.
Per capirne il funzionamento, bisogna far riferimento al modo in cui opera il VIC all'interno del Commodore 64.
Esso trasferisce al televisore, linea per linea, le informazioni di cui questo ha bisogno per ricostruire l'immagine, andando a prenderle nella memoria con un metodo di indirizzamento che dipende dal modo grafico in cui si trova.

Il VIC contiene un contatore (raster) il cui contenuto indica la linea di scansione attuale.
Per avere la parte alta dello schermo in un modo grafico e la parte bassa in un altro, bisogna cambiare modo grafico in corrispondenza di due determinati valori di questo contatore.
Il VIC permette di fare questo; infatti uno dei suoi registri, il raster register, registro numero 18, contiene sempre il valore del raster; inoltre, scrivendo un certo valore in questo registro, si può programmare il VIC

Listato 2. *Tavola dei simboli.*

```
 53248 : $D000  VIC
56320 : $DC00  CIA1
56576 : $DD00  CIA2
788 :   $0314  CINV
655 :   $028F  KEYLOG
59953 : $EA31  IRQK
203 :   $00CB  SFDX
197 :   $00C5  LSTX
653 :   $028D  SHFLAG
32768 : $8000  BANK
32770 : $8002  MEMPTS
32772 : $8004  RASTER
32774 : $8006  DELAY
32775 : $8007  FREQ
32776 : $8008  SWAP
32786 : $8012  A1
32834 : $8042  IRQHIR
32844 : $804C  SYNC
32846 : $804E  B2
32854 : $8056  B1
32857 : $8059  B3
32863 : $805F  HIRON
32888 : $8078  HIROFF
32914 : $8092  C1
32920 : $8098  UP
32932 : $80A4  D1
32936 : $80A8  DOWN
32946 : $80B2  D2
32950 : $80B6  KEYON
32964 : $80C4  KEYOFF
32978 : $80D2  UDKEY
32998 : $80E6  E1
33001 : $80E9  EXEC
33028 : $8104  F3
33034 : $810A  F1
33055 : $811F  F4
33058 : $8122  F2
```

Alta risoluzione

Listato 3. *Codici da inserire a partire dalla locazione 32768. Ogni linea contiene otto codici. Ogni quattro linee vi è un codice di controllo uguale alla somma dei 32 codici precedenti. Il lettore dovrebbe aggiungere il programma caricatore in BASIC.*

```
 100 DATA2,3,120,16,152,0,16,1
110 DATA162,0,173,17,208,41,32,240
120 DATA1,232,189,4,128,141,18,208
130 DATA173,17,208,41,127,73,32,141
140 DATA2916
150 DATA17,208,173,24,208,41,7,29
160 DATA2,128,141,24,208,173,2,221
170 DATA9,3,141,2,221,173,0,221
180 DATA41,252,29,0,128,141,0,221
190 DATA3188
200 DATA202,96,173,25,208,41,1,240
210 DATA13,141,25,208,162,5,202,16
220 DATA253,32,8,128,240,3,76,49
230 DATA234,104,168,104,170,104,64,8
240 DATA3503
250 DATA120,169,66,141,20,3,169,128
260 DATA141,21,3,169,1,141,13,220
270 DATA13,26,208,141,26,208,40,96
280 DATA8,120,169,49,141,20,3,169
290 DATA2962
300 DATA234,141,21,3,173,26,208,41
310 DATA254,141,26,208,169,129,141,13
320 DATA220,40,32,8,128,208,251,96
330 DATA173,4,128,56,233,8,201,48
340 DATA3762
350 DATA176,2,169,48,141,4,128,96
360 DATA173,4,128,24,105,8,144,2
370 DATA169,248,141,4,128,96,8,120
380 DATA169,210,141,143,2,169,128,141
390 DATA3369
400 DATA144,2,40,96,8,120,169,72
410 DATA141,143,2,169,235,141,144,2
420 DATA40,96,173,141,2,201,4,208
430 DATA13,165,203,201,3,144,7,201
440 DATA3430
450 DATA7,176,3,32,233,128,76,72
460 DATA235,170,197,197,208,28,41,2
470 DATA240,48,206,6,128,208,43,238
480 DATA6,128,206,7,128,208,35,169
490 DATA3809
500 DATA4,141,7,128,138,106,144,144
510 DATA176,158,160,16,140,6,128,160
520 DATA1,140,7,128,41,2,208,236
530 DATA138,106,144,3,76,95,128,76
540 DATA3285
550 DATA120,128,96
560 DATA-1,344
```

perché richieda un interrupt quando il contatore uguaglia quel valore.
Se la routine di interrupt, oltre alle sue funzioni normali, si occupa anche di cambiare modo grafico quando la richiesta è fatta dal VIC, il gioco è fatto.
Bisogna osservare però che se per qualche motivo alcune richieste di interrupt da parte del VIC andassero perdute, si avrebbero conseguenze spiacevoli sullo schermo.

Ora, questo è proprio quello che succede se vi è anche un'altra sorgente di interrupt, poiché mentre il calcolatore esegue la routine corrispondente, non accetta eventuali altre richieste.
Normalmente il Commodore 64 usa come sorgente di interrupt un timer contenuto nella CIA (Complex Interface Adapter) numero 1, programmato per produrre 60 richieste di interrupt al secondo.
Per quanto detto sopra, si rende necessario disabilitare questo dispositivo, e usare il raster come unica sorgente di interrupt.
Fortunatamente ciò è possibile, poiché la frequenza del raster, uguale a quella televisiva di semiquadro (50 Hz), differisce poco dai 60 Hz del timer.
L'unica differenza che si nota è un leggero rallentamento nel lampeggio del cursore; inoltre la variabile BASIC TI contiene il tempo in cinquantesimi di secondo anziché in sessantesimi.
Irqhir è la nuova routine di interrupt, e Hiron e Hiroff, oltre a modificare il vettore in $314-315, selezionano la sorgente di interrupt.
Passiamo alla seconda funzione del programma, il riconoscimento dei tasti funzione.
La routine è Udkey, che viene attivata e disattivata rispettivamente da Keyon e Keyoff.

Su Udkey non c'è molto da dire: essa esamina il tasto premuto, e agisce di conseguenza usando Hiron e Hiroff per accendere e spengere la pagina grafica, e Up e Down per diminuirne o aumentarne la parte visibile.

I salti indiretti in $314-315 e in $28F-290

Molte funzioni del sistema operativo del Commodore 64 sono vettorizzate, sono cioé eseguite da routine che vengono richiamate tramite un salto indiretto ad una locazione di memoria RAM.
Ciò rende il sistema molto flessibile perché, modificando il vettore in RAM, si può sostituire, in tutto o in parte, la routine corrispondente.
I vettori che ci interessano sono quelli in $314-315 e in $28F-290.
$314-315 è relativo all'interrupt, e il suo valore normale ($EA31) punta al programma che gestisce il lampeggio del cursore, l'aggiornamento dell'orologio in sessantesimi e la scansione della tastiera.
All'interno della routine di scansione tastiera, l'esame del tasto premuto è a sua volta vettorizzato in $28F-290.
Il valore normale ($EB48) di questo vettore punta al programma che si occupa principalmente di trovarne il codice ASCII e di immetterlo nel buffer di tastiera.

Questo programma ha come ingresso le locazioni di memoria $CB, che contiene il codice del tasto premuto (numero legato alla sua posizione sulla tastiera, non codice ASCII), e $28D che contiene lo stato dei tasti SHIFT (bit 0), Commodore (bit 1) e CONTROL (bit 2). ■

LIBRERIA

JACKSON

A Milano, in via Mascheroni, 14.
La prima software - libreria italiana

Un tempo si andava in libreria per il gusto della scoperta, per il piacere di esser informati sulle novità. Per incontrarsi, discutere, chiedere un consiglio al libraio-amico. Tutto questo è ancora possibile, per un prodotto assolutamente nuovo: libri e riviste di informatica italiani ed esteri, software, giochi.

Dove? Alla **Libreria JACKSON**. La prima software - libreria italiana.

A Milano, in via Mascheroni, 14. – Tel. 02-437385

Vieni a trovarci: ti aspettiamo.

PERSONAL SOFTWARE

Grafica e joystick per Apple II

Due utility molto interessanti

di *Claudio Poma*

Spesso quando si parla di un computer gli argomenti principe sono il numero di bit del processore e i Kbyte di memoria. Si tratta senz'altro degli aspetti più rilevanti del problema, ma ne esiste un altro che, seppur taciuto, non ha minore importanza dei precedenti ed è il buon uso del sistema posseduto e se i primi sono strettamente legati all'hardware, e quindi poco manipolabili, il secondo dipende prevalentemente dal software ed è quello sul quale l'utente finale può lavorare con i migliori risultati.

In riferimento a quanto appena detto, ora tratteremo alcuni aspetti della grafica e dell'uso dei joystick nell'Apple II.

Le tavole delle figure

Tutti i possessori di un Apple II sanno che la grafica ad alta risoluzione della loro macchina è gestibile in due modi: da una parte si possono tracciare i disegni desiderati facendo uso della istruzione HPLOT, dall'altra si possono approntare delle tavole di figure da utilizzare al momento opportuno e con degli ordini predisposti.

La prima è senz'altro la più semplice da gestire e va benissimo se si devono tracciare grafici o istogrammi, ma se si tenta di creare un movimento di immagini i problemi crescono a causa di una lentezza non sempre accettabile. La soluzione ci è data dalla seconda opzione prima menzionata, ma qui ci sono i primi dolori. Se i risultati sono buoni, l'ottenerli non è la cosa più chiara a meno di non far ricorso a dei programmi predisposti, ma che, oltre all'essere in genere costosi, non sempre valgono la candela, soprattutto se il disegno richiesto non è molto complesso.

Vediamo, quindi, come definire delle figure e come usarle in programmi di movimento.

Per prima cosa ci si deve premunire di fogli a quadretti sui quali tracciare i contorni che delimitino le dimensioni massime dell'immagine che vogliamo rappresentare. Eseguita la cornice è la volta del disegno vero e proprio e, una volta

Listato 1. *Il programma per l'aiuto grafico sull'Apple.*

```
TAVOLE DELLE FORME

10  DIM S1(500),V1(500),VE$(500)
20  POKE 34,0: HOME
25  INVERSE : PRINT "MENU'  TAVOLA DELLE
 FORME": NORMAL
30  PRINT : PRINT : PRINT "1)  Nuova tav
ola"
33  PRINT : PRINT "2)  Cambiamento vetto
ri"
35  PRINT : PRINT "3)  Visualizzazione v
ettori"
37  PRINT : PRINT "4)  Conversione vetto
ri"
39  PRINT : PRINT "5)  FINE"
41  PRINT : PRINT : PRINT : PRINT "Dimmi
 un numero e premi  <RETURN> ";: INPUT
     W
43  IF W < 1 OR W > 5 THEN 20
44  HOME
45  ON W GOTO 50,75,500,100,450
50 M$ = "":I = 0
55 V = I: GOSUB 270
60  IF M$ <  > "E" THEN S1(I) = M:I = I
+ 1: GOTO 55
65  POKE  - 16368,10
70  GET A$: GOTO 20
75  POKE  - 16368,10: PRINT
80  INPUT "Vettore da cambiare (0=FINE)
: ";V
85  IF V < 0 THEN 75
90  IF V > 0 THEN V = V - 1: GOSUB 270:S
1(V) = M: GOTO 75
95  GOTO 20
100  FOR V = 0 TO I
110  IF B = 2 AND S1(V) > 0 AND S1(V) <
4 THEN 140
120  IF B < 2 AND (S1(V) > 0 OR S1(V) >
4) THEN 140
130 B = 0:Q = Q + 1
140 V1(Q) = V1(Q) + S1(V) * (8 ^ B)
150 B = B + 1
160  IF B > 2 THEN B = 0:Q = Q + 1
170  NEXT
180  PRINT "Byte";: HTAB 15: PRINT "ESA.
";: HTAB 27: PRINT "DEC."
185  PRINT : POKE 34,2
190  FOR V = 0 TO Q
195  IF (V / 15) =  INT (V / 15) THEN  G
ET A$
200 H% = V1(V) / 16
210 L% = V1(V) - H% * 16
```

ultimatolo, bisogna segnare tutti i quadretti di contorno fino all'ottenimento della forma stilizzata dell'oggetto in questione. A questo punto, partendo da un punto della figura si deve descrivere il disegno usando i simboli: su, giù, destra, sinistra, e per ognuno si deve segnalare se con o senza tracciamento (cioé se il quadretto in questione fa parte o no della forma stilizzata).
Attenzione perché non si può andare in diagonale e, altra regola molto importante, mai usare per due o più volte di seguito la funzione Su senza tracciare.
La prossima operazione richiesta per raggiungere la sospirata visione del movimento dell'immagine prescelta è la conversione, dell'insieme dei codici binari dei simboli appena raccolti, in numeri decimali o esadecimali e si tratta della fase più delicata perché una minima distrazione può costare tutto il lavoro fin qui svolto, ma non spaventatevi, non dovrete lasciare gli occhi su delle interminabili tavole di conversione purché abbiate la pazienza di copiarvi il listato 1.
Questo, libero adattamento del programma pubblicato a pag. 227 del libro *Apple II guida all'uso*, del Gruppo Ed. Jackson, oltre a correggere alcune sviste del listato appena menzionato, ne permettono un utilizzo più semplice e chiaro, come si può facilmente desumere dalle linee che riportano il menù.

L'inserimento dei dati avviene per GET e non più tramite INPUT, il che velocizza l'operazione, ma questa non presenta particolari rischi perché la visualizzazione è gestita dal computer (linee 330-430) che per ogni ordine trascrive la funzione completa e non il solo tasto premuto.
Chi non avesse l'uso delle minuscole può adattare il programma cambiando opportunamente i valori di K alle linee 370-400 con i codici ASCII dei rispettivi CTRL. I tasti usati, a proposito, sono: I-i per Su, J-j per Sinistra, L-l per Destra, M-m per Giù, le maiuscole con tracciamento, le minuscole senza.
Il programma permette, inoltre, una visualizzazione ripetuta sia dei dati immessi, per un eventuale controllo, che delle conversioni in esadecimale e decimale. Lo scroll di questi dati, nel caso di eccessiva lunghezza, è gestito dal computer che ne presenta quindici alla volta (linee 195 e 540) attendendo, per continuare, la pressione di un tasto.
L'aggiunta, a fianco della esadecimale, della conversione in valori de-

```
220  IF H% > 9 THEN H% = H% + 7
230  IF L% > 9 THEN L% = L% + 7
240  HTAB (5 - LEN ( STR$ (V))): PRINT
V;: HTAB 16: PRINT CHR$ (H% + 176
     ); CHR$ (L% + 176);: HTAB (30 - LE
N ( STR$ (V1(V)))): PRINT V1(V)
245 V1(V) = 0
250  NEXT
253 B = 0
255 Q = 0: POKE - 16368,10
260  PRINT : PRINT "Per continuare premi
 un tasto ";: GET A$: GOTO 20
270  PRINT "IJLM ijlm   ";
280  PRINT "VETTORE ";V + 1" ";
290  POKE - 16368,10
300  IF PEEK ( - 16384) < 128 THEN 300
310 K = PEEK ( - 16384)
320 M = 9
330  IF K = 201 THEN M = 4:VE$(V) = "SU
+"
340  IF K = 204 THEN M = 5:VE$(V) = "DES
+"
350  IF K = 205 THEN M = 6:VE$(V) = "GIU
+"
360  IF K = 202 THEN M = 7:VE$(V) = "SIN
+"
```

```
370  IF K = 233 THEN M = 0:VE$(V) = "SU
-"
380  IF K = 236 THEN M = 1:VE$(V) = "DES
-"
390  IF K = 237 THEN M = 2:VE$(V) = "GIU
-"
400  IF K = 234 THEN M = 3:VE$(V) = "SIN
-"
410  IF K = 197 OR K = 229 THEN M =  - 1
:M$ = "E":VE$(I) = "FINE."
420  IF M = 9 THEN 290
425  IF M =  - 1 THEN S1(I) = 0
430  PRINT VE$(V)
440  RETURN
450  HOME : END
500  HOME
510  PRINT " N.      Vettore"
520  PRINT : POKE 34,2
530  FOR T = 0 TO I
540  IF (T / 15) =  INT (T / 15) THEN  G
ET A$
550  PRINT T;: HTAB 10: PRINT VE$(T)
560  NEXT
570  POKE - 16368,10
580  PRINT : PRINT "Per continuare premi
 un tasto ";: GET A$: GOTO 20
```

cimali dei valori binari dei vettori grafici non è stata inserita a caso nel programma, ma ci introduce alla prossima fase, forse la più oscura, che riguarda l'inserimento, della tavola appena ottenuta, in un programma di movimento.

Per poter essere adeguatamente sfruttata, la lista di numeri che abbiamo appena trovato con l'ausilio del programma riportato nel listato N. 1 deve essere allocata in una qualche zona della memoria RAM, e per far questo abbiamo due possibilità: la prima prevede il richiamo del monitor battendo CALL-151 e l'inserimento, nella zona prescelta, dei valori esadecimali appena ottenuti, che dovranno essere registrati su dischetto con un BSAVE e quindi richiamati con un BLOAD, la seconda richiede l'uso dell'istruzione POKE che, direttamente dal programma interessato all'utilizzo della figura, carica in memoria i valori decimali. La prima opzione è senz'altro quella stilisticamente più "professionale", ma non deve essere sempre così sicura. Se, da una parte, nell'utilizzare il monitor c'è il vantaggio di poter richiamare in programmi diversi la stessa e complessa figura senza doverla riscrivere ogni volta, dall'altra esiste il problema della Directory che deve sempre precedere la tavola delle forme e che contiene i dati riguardanti il numero delle figure e il byte di inizio di ogni figura (cioè la distanza del primo byte dall'inizio della directory) che è chiamato puntatore e il cui valore è suddiviso in due byte. Nella compilazione della directory il byte di ordine basso deve sempre precedere quello di ordine alto (a chiarimento vedere figura 1).

È proprio sul ruolo di questi puntatori che si gioca la scelta sull'uso del monitor o delle POKE per l'inserimento dei dati della tavola delle forme. Se la prima possibilità è formalmente la più corretta, presenta però lo svantaggio di non essere modificabile, cioè di non permettere, a meno di una illuminata lungimiranza da parte del programmatore, l'inserimento, in tempi successivi, di altre figure. Questo è invece facilissimo se si fa uso delle POKE che, per contro, richiedono la trascrizione dei dati in ogni programma.

L'utilizzo di un metodo a danno dell'altro deve quindi considerare la lunghezza dei dati e gli eventuali sviluppi futuri che, in ogni caso, nell'uso del monitor possono essere prevenuti inserendo nella directory dei byte vuoti (vedi figura 1).

Byte	Esa.	Dec.	Descrizione
0	02	02	Numero di figure nella tavola
1	00	00	Byte mai usato
2	08	8	Byte basso puntatore figura 1
3	00	0	Byte alto puntatore figura 1
4	0C	12	Byte basso puntatore figura 2
5	00	0	Byte alto puntatore figura 2
6	00	0	Byte basso puntatore futura figura 3
7	00	0	Byte alto puntatore futura figura 3
			Fine Directory
8	A1	161	Inizio figura 1
9	5C	92	
10	09	9	
11	00	0	Fine figura 1
12	B1	177	Inizio figura 2
13	65	101	
14	3A	58	
15	00	0	Fine figura 2

Figura 1. *Esempio di organizzazione di una tavola delle forme comprendente due brevi figure e con lo spazio, nella Directory, per il futuro inserimento di una terza. Per le differenze nell'uso dei valori esadecimali e decimali si rimanda all'articolo.*

The Race

Ecco un esempio di quanto appena detto, una vera e propria corsa di cavalli con tanto di scommesse, quotazioni e immancabile gara tra sei cavalli e relativi cavalieri, il tutto ad una velocità degna dei migliori videogiochi pur essendo il programma scritto completamente in BASIC.

In questa occasione abbiamo optato sulla seconda opzione summenzionata e all'inizio del listato 2 troviamo il ciclo che dà forma alla figura del cavallo prendendo gli ordini necessari dai DATA delle linee da 80 a 140, quindi si ha una prima presentazione del disegno ingrandito 3 volte.

A questo punto si deve informare il computer sul numero e nome delle persone che intendono partecipare alla prova e ultimata questa fase si passa al gioco.

Ogni giocatore deve fare le sue puntate su quotazioni che vengono scelte a caso di volta in volta e con l'unica limitazione di non puntare più di 249 dollari su un singolo cavallo, oltre, naturalmente, a non spendere più di quanto si possiede.

Quando tutti hanno effettuato le loro giocate si arriva alla fase più appassionante, anche per il non indifferente effetto grafico, e qui come non mai è opportuno richiamare il discorso iniziale sull'importanza di saper usufruire di tutte le possibilità offerte dalla macchina che si sta usando. Basta veder una sola corsa per capire come il discorso sulla validità o meno di un computer non sia legato solo alle capacità del processore, ma anche al grado di utilizzo dello stesso. Tutto questo per dire che se l'impegnarsi nella comprensione dell'argomento fin qui trattato può richiedere un certo sforzo, il

Listato 2. *Questo programma simula una corsa di cavalli.*

```
5  SCALE= 1
10  HOME
20  VTAB 11: HTAB 17: FLASH : PRINT "CAV
ALLI": NORMAL
40  POKE 232,0: POKE 233,3
50  FOR X = 768 TO 768 + 140
60  READ Y: POKE X,Y
70  NEXT
80  DATA 01,00,04,00,118,36,36,44,45,44,
45,44,13,44
85  DATA 45,53,182,45,45,45,13,36,36,39
90  DATA 28,100,44,53,174,62,77,44,252,2
8
95  DATA 36,12,12,12,252,252,22,191,191,
23
100  DATA 23,183,21,53,62,63,45,77,62,54
105  DATA 54,213,27,216,27,216,3,24,24,2
3
110  DATA 181,55,54,30,30,30,246,50,54,6
1
115  DATA 100,36,108,12,108,108,12,12,54
,54
120  DATA 46,182,149,45,56,56,36,36,103,
12
125  DATA 12,12,46,45,109,22,21,53,45,46
130  DATA 150,10,45,248,36,36,12,36,60,2
14
135  DATA 26,231,12,44,5,56,100,100,12,1
2
140  DATA 36,21,21,13,44,38,28,28,28,28,
228,62,156,62,31,6,0,0
142  HGR2 : SCALE= 3
143  DRAW 1 AT 80,70
144  FOR T = 1 TO 6000: NEXT : SCALE= 1
145  TEXT
150 C$ = "    "
170 R1$ = "-----------|---|---|---|---|-
---|---|----"
180 R2$ = "|    |    |    |    |    |    |"
190 VA$(1) = "3/2":VA$(2) = "2/1":VA$(3)
 = "5/2":VA$(4) = "3/1":VA$(5) = "
    7/2":VA$(6) = "4/1"
195 VA(1) = 1.5:VA(2) = 2:VA(3) = 2.5:VA
(4) = 3:VA(5) = 3.5:VA(6) = 4
200  HOME
210  INPUT "In quanti giochiamo ? (max.7
) ";GI
220  IF GI < 1 OR GI > 7 THEN 200
230  PRINT : FOR T = 1 TO GI
240  PRINT : PRINT "Nome del giocatore N
."T;: INPUT " ";GI$(T)
250  IF  LEN (GI$(T)) > 11 THEN  PRINT "
NOME TROPPO LUNGO. Max.11 lettere.
     ": GOTO 240
260  NEXT
270  PRINT : PRINT "A questa partita gio
cheranno :"
280  FOR T = 1 TO GI: PRINT GI$(T): NEXT
290  PRINT : PRINT "VA tutto bene ? (S/N
) ";
300  GET A$: IF A$ = "N" THEN 200
310  IF A$ <  > "S" THEN 300
320 PO = 7 + GI * 2: IF PO > 19 THEN PO
= 19
330  FOR T = 1 TO 7:SO(T) = 1000: NEXT
400  FOR T = 1 TO 6
410 CA =  INT ( RND (1) * 6) + 1
420 QU$(T) = VA$(CA)
430 QU(T) = VA(CA)
440  NEXT
500  HOME
510  PRINT R1$;
520  PRINT "  Cavallo  | 1 | 2 | 3 | 4 |
 5 | 6 |"
530  PRINT R1$;
540  PRINT "Quotazioni |";: FOR T = 1 TO
 6: PRINT QU$(T)"|";: NEXT
550  PRINT "($$)";
560  PRINT R1$;
570  FOR T = 1 TO GI
580  PRINT GI$(T);: HTAB 12: PRINT R2$;:
 HTAB (41 -  LEN ( STR$ (SO(T)))):
      PRINT SO(T);
590  PRINT R1$;
600  NEXT
610  FOR G = 1 TO GI
615  IF SO(G) <  = 0 THEN 780
620  VTAB PO: HTAB 1: CALL  - 868
630  VTAB PO: PRINT "Tocca a "GI$(G)
640  POKE 34,PO + 1: CALL  - 198
650  FOR C = 1 TO 6
660  VTAB PO + 2: HTAB 1: CALL  - 868
670  VTAB PO + 2: PRINT "Puntata per il
cavallo N."C;: INPUT " ";SC$(C,G)
675  IF  VAL (SC$(C,G)) > 249 THEN 660
680  NEXT C
690 SO = 0: FOR T = 1 TO 6:SO = SO +  VA
L (SC$(T,G)): NEXT
700  IF SO > SO(G) THEN  PRINT GI$(G)" h
ai scommesso troppo!!!": FOR T = 1
     TO 2000: NEXT : GOTO 650
710 SO(G) = SO(G) - SO
720  POKE 34,0
730  VTAB 4 + 2 * G
740  FOR T = 16 TO 36 STEP 4
750  HTAB (T -  LEN (SC$(T / 4 - 3,G))):
 PRINT SC$(T / 4 - 3,G);
760  NEXT
765  HTAB 37: PRINT C$;
770  HTAB (41 -  LEN ( STR$ (SO(G)))): P
RINT SO(G);
780  NEXT G
790  VTAB PO + 4: PRINT "I giochi sono f
atti.": FOR T = 1 TO 2000: NEXT : HTAB
     22: PRINT "Ecco la corsa."
800  FOR T = 1 TO 3000: NEXT
810  HOME
820  HGR2
830  HCOLOR= 1
840  FOR T = 29 TO 174 STEP 29
850  HPLOT 1,T TO 279,T
860  NEXT
870  HCOLOR= 6
```

non farlo significa perdere l'opportunità di riuscire a raggiungere la capacità di sfruttare completamente il piccolo gioiello in nostro possesso, con le conseguenze del caso.

Per mantenere una certa incertezza sul risultato si è legato lo spostamento dei disegni ad una decisione casuale del computer, linea 1000, anche se il differente peso delle quotazioni è in parte mantenuto dalla lunghezza di ogni singolo spostamento, linea 900, maggiore in rapporto a superiori possibilità di vittoria presupposte.
Terminata la corsa si viene informati circa il numero del cavallo vincitore e di seguito viene riportata la quotazione dello stesso per quella prova con le eventuali vincite di qualche concorrente, quindi si passa ad una nuova gara con nuove quotazioni.

Joystick e interattività

Il prossimo argomento è senz'altro meno complicato da comprendere, ma non per questo con minori possibilità di utilizzo. L'Apple II ha una precisa funzione per l'uso di paddle: PDL(espr.), dove espr può assumere i valori da 0 a 3. In pratica ciò permette la gestione di 4 controllori di giochi. Se usate un paddle, il valore ritornato dall'uso della funzione è compreso tra 0 e 255, mentre nel caso di impiego di un joystick la condizione è un po' più complessa, ma, come vedremo, non troppo.
Prima, però, un accenno sul valore di un simile intervento.
Nell'impostare i programmi, a volte non si tiene debitamente conto del bisogno umano di sviluppare i rapporti relazionali nel modo più semplice e chiaro.
Sull'onda dell'importanza di questo aspetto si è giunti all'attuale dilagare di mouse e affini e i joystick, parenti, in parte poveri, di questi, possono, in impieghi non al limite, farne convenientemente le veci. Inoltre l'Applesoft ci permette di utilizzarne due e questo è molto importante, come vedremo, nei giochi di azione che prevedono la partecipazione di due giocatori dove l'uso della tastiera, pur permettendone lo svolgimento, ne limita le possibilità per la particolare gestione della stessa.

Tornando alla funzione PDL(x), abbiamo già detto sul suo comportamento durante l'impiego di un paddle. La prima differenza coi joystick è che uno di questi impiega due delle quattro opzioni a disposizione, la seconda è che i valori ottenibili sono fissi. Per un quadro riepilogativo si veda la figura 2, mentre per il lato pratico si giri il breve programma del listato 3 che ci permette di visualizzare quanto appena detto.
In genere i joystick permettono otto movimenti cioé, oltre a: su, giù, destra e sinistra, anche le diagonali. Il programma appena citato fa luce anche su questo aspetto; in definitiva si ottengono i valori dei due punti cardinali interessati. Quando nessuna direzione è scelta, i valori riportati dalle funzioni PDL si mantengono intorno a valori centrali del periodo 0-255 pur con alcune minime variazioni ininfluenti, però, sull'aspetto analizzato.
A questo punto introdurre le nozioni apprese nei programmi più disparati non dovrebbe presentare insormontabili difficoltà.

Trappola

Per tradurre in pratica quanto fin qui detto, ecco una versione di uno

Seguito listato 2.

```
880  HPLOT 260,1 TO 260,174
885 FO = 0
890  HCOLOR= 3: FOR K = 1 TO 6
900 FO(K) = 25 - QU(K) * 4 -  INT ( RND
(1) * 4)
910 Y(K) = 18 + 29 * (K - 1):X(K) = 1
920  DRAW 1 AT X(K),Y(K)
930  NEXT
1000 K =  INT ( RND (1) * 6) + 1
1010  FOR T = 1 TO 20: NEXT
1020  XDRAW 1 AT X(K),Y(K)
1030 W =  PEEK ( - 16336)
1040 X(K) = X(K) + FO(K)
1050  DRAW 1 AT X(K),Y(K)
1060  IF X(K) > 220 THEN 1100
1070  FOR R = 1 TO 3:W =  PEEK ( - 16336
): NEXT
1090  GOTO 1000
1100  VTAB 21: PRINT "Ha vinto il cavall
o N."K
1110  FOR T = 1 TO 4000: NEXT
1120  TEXT : HOME
```

```
1130  PRINT "Ha vinto il cavallo N."K
1140 B$ = "--------------------------"
1150 D$ = "|-----------|---|---|----|"
1160  PRINT : PRINT "------------------"
1170  PRINT "|Quotazione |"QU$(K)"|": PR
INT B$
1180  PRINT "|Puntata di      |(+)|($$)|"
1190  FOR T = 1 TO GI: PRINT D$
1200 SO(T) = SO(T) +  INT ( VAL (SC$(K,T
)) * QU(K))
1210  PRINT "|"GI$(T);: HTAB 13: PRINT "
|";: HTAB (17 -  LEN (SC$(K,T))): PRINT
     SC$(K,T)"|";
1220  HTAB (21 -  LEN ( STR$ ( VAL (SC$(
K,T)) * QU(K)))): PRINT  VAL (SC$(
     K,T)) * QU(K)"|";: HTAB (26 -  LEN
( STR$ (SO(T)))): PRINT SO(T)"|"
1230  NEXT
1240  PRINT B$
1250  PRINT : PRINT : PRINT "Per un'altr
a corsa premere un tasto ";
1260  POKE  - 16368,12
1270  GET A$: GOTO 400
```

Listato 3. *Il programma "Joystick e interattività".*

```
10  HOME
20  PRINT "PDL      PDL      PDL      PDL"
30  PRINT "(0)      (1)      (2)      (3)"
40  POKE 34,2
50  HOME
60  PRINT  PDL (0);: HTAB 9: PRINT  PDL
(1);: HTAB 17: PRINT  PDL (2);: HTAB
    25: PRINT  PDL (3)
70  FOR T = 1 TO 200: NEXT
100  GOTO 50
```

Listato 4. *Questo gioco traduce quanto detto in precedenza.*

TRAPPOLA

```
10  HOME
20  VTAB 10: HTAB 17: FLASH : PRINT "TRA
PPOLA": NORMAL
30  FOR T = 1 TO 5000: NEXT
40  TEXT : HOME
50  VTAB 3: INPUT "Nome giocatore di DES
TRA ";GD$
60  VTAB 5: INPUT "Nome giocatore di SIN
ISTRA ";GS$
70  VTAB 8: PRINT "A questa partita gioc
heranno :"
80  VTAB 10: PRINT GD$: PRINT GS$
90  VTAB 13: PRINT "Va tutto bene ? (S/N
) ";
100  GET A$: IF A$ = "N" THEN 40
```

```
110  IF A$ <  > "S" THEN 100
120  PRINT : PRINT : INPUT "Quanti ruond
 deve durare la partita? ";NP
130  PRINT : PRINT "A che velocita' vole
te giocare ?"
140  PRINT : PRINT "1- Veloce"
150  PRINT "2- Media"
160  PRINT "3- Lenta"
170  PRINT : INPUT "Dimmi il numero ";VE
180  IF VE < 1 OR VE > 3 THEN 170
190  PRINT : PRINT "Premere un tasto per
 iniziare la gara ";
200  GET A$
210 PD = 0:PS = 0
220 NU = 0
230  HOME
300  GR
310  COLOR= 13
320  FOR X = 0 TO 39:Y = 0: PLOT X,Y:Y =
 39: PLOT X,Y: NEXT
330  FOR Y = 0 TO 39:X = 0: PLOT X,Y:X =
 39: PLOT X,Y: NEXT
340 SI$ = GS$ + " - " + GD$ + "  " +  ST
R$ (PS) + "-" +  STR$ (PD)
350  VTAB 21: HTAB ((44 -  LEN (SI$)) /
2): PRINT SI$
360  IF NU = NP THEN 800
370 DD = 4:DS = 4
380  VTAB 21: PRINT NU + 1"su"NP
390 XD = 30:YD = 20:XS = 10:YS = 20:CD =
 6:CS = 9
400  COLOR= CD: PLOT XD,YD
410  COLOR= CS: PLOT XS,YS
420  IF  PDL (0) = 255 THEN DD = 1: GOTO
 460
```

Joystick 2	Posizione	Joystick 1
PDL(3) 0	Su	PDL(1) 0
PDL(3) 255	Giù	PDL(1) 255
PDL(2) 255	Destra	PDL(0) 255
PDL(2) 0	Sinistra	PDL(0) 0

Figura 2. *Tabella riassuntiva dei valori ottenibili utilizzando la funzione PDL per la gestione dei joystick.*

dei più classici giochi per due giocatori.
Ogni concorrente deve manovrare un punto che non può stare fermo e che lascia dietro di sé una vistosa traccia. La gara termina quando uno dei due punti tocca o un tracciato o il bordo del campo e il vincitore è, naturalmente, chi ha resistito di più.

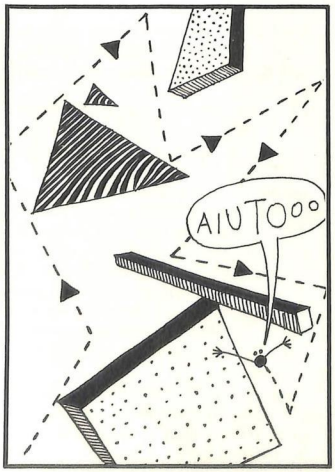
L'uso dei joystick ha permesso di superare quello che era il punto dolente delle versioni giocate con la tastiera e, cioé, il fatto che, essendo il computer in grado di leggere un solo tasto alla volta, la partita poteva essere facilmente "truccata", anche da gesti del tutto involontari, tramite la pressione continua di un tasto.
Ora, invece, ogni concorrente ha la possibilità di manovrare liberamente il proprio punto senza interferire, o poter interferire, sulla tattica dell'avversario. Il programma risulta quindi più giocabile ed avvincente, anche perché sono previste varie opzioni che garantiscono sia vari livelli di gioco che una chiara e aggiornata informazione sugli avvenimenti in corso.
Dopo l'inserimento e il controllo dei nomi degli sfidanti si deve decidere su quanti round si deciderà la gara, quindi è la volta della velocità del gioco.
Inizia la partita. Disegnato il campo (linee 320-330), visualizzato il punteggio (340-350) e il numero del round che si sta giocando (380), i simboli iniziano a muoversi fino a quando uno dei due toccherà il bordo del campo o uno dei percorsi. All'inizio il movimento di entrambi i giocatori è programmato verso il basso (370), ma ogni cambiamento in tal senso non presenta alcun problema. La linea 390 definisce le coordinate di partenza e i colori dei giocatori.

Le linee da 420 a 570 sono, però, quelle che maggiormente ci interessano per l'argomento trattato in queste pagine.

Seguito listato 4.

```
430  IF  PDL (0) = 0 THEN DD = 2: GOTO 4
60
440  IF  PDL (1) = 0 THEN DD = 3: GOTO 4
60
450  IF  PDL (1) = 255 THEN DD = 4
460  IF  PDL (2) = 255 THEN DS = 1: GOTO
 500
470  IF  PDL (2) = 0 THEN DS = 2: GOTO 5
00
480  IF  PDL (3) = 0 THEN DS = 3: GOTO 5
00
490  IF  PDL (3) = 255 THEN DS = 4
500  IF DD = 1 THEN XD = XD + 1: GOTO 54
0
510  IF DD = 2 THEN XD = XD - 1: GOTO 54
0
520  IF DD = 3 THEN YD = YD - 1: GOTO 54
0
530  IF DD = 4 THEN YD = YD + 1
540  IF DS = 1 THEN XS = XS + 1: GOTO 60
0
550  IF DS = 2 THEN XS = XS - 1: GOTO 60
0
560  IF DS = 3 THEN YS = YS - 1: GOTO 60
0
570  IF DS = 4 THEN YS = YS + 1
600  IF  SCRN( XD,YD) <  > 0 OR  SCRN( X
S,YS) <  > 0 THEN  CALL  - 198: GOTO
     700
610  FOR T = 1 TO 50 * VE ^ 2: NEXT
620  FOR T = 1 TO 3:A =  PEEK ( - 16336)
: NEXT
```

```
650  GOTO 400
700  IF  SCRN( XD,YD) = 0 OR  SCRN( XS,Y
S) = 0 THEN 740
710  VTAB 23: PRINT "Il round e' finito
in parita'."
720 PD = PD + 1:PS = PS + 1
730  GOTO 760
740 A$ = GD$:PD = PD + 1: IF  SCRN( XS,Y
S) = 0 THEN A$ = GS$:PD = PD - 1:P
     S = PS + 1
750  VTAB 23: PRINT "Il round e' stato v
into da "A$
760 NU = NU + 1
770  FOR T = 1 TO 3000: NEXT : IF NU = N
P THEN  HOME : GOTO 340
780  PRINT "Premi un tasto per il prossi
mo round. ";
790  GET A$: HOME : GOTO 300
800 A$ = GD$: IF PS > PD THEN A$ = GS$
805  IF PD = PS THEN  VTAB 23: PRINT "La
 partita e' terminata in parita'."
     : GOTO 820
810  VTAB 23: PRINT "La partita e' stata
 vinta da ";: FLASH : PRINT A$: NORMAL

820  FOR T = 1 TO 3000: NEXT : VTAB 24:
HTAB 15: PRINT "Ancora ? (S/N) ";
830  GET A$: IF A$ = "N" THEN  TEXT : HO
ME : END
840  IF A$ <  > "S" THEN 830
850  PRINT : PRINT "Giocate alla stesse
condizioni ? (S/N) ";
860  GET A$: IF A$ = "N" THEN 40
870  IF A$ <  > "S" THEN 860
880  GOTO 190
```



La vista di ben 16 IF THEN consecutivi farà inorridire più di un lettore, e a ragione visto che il tutto può essere formulato in una forma più concisa ed elegante, ma si tenga conto dell'intenzione dichiarata di presentare un esempio il più chiaro possibile anche a dei neofiti. Per questi, DD e DS sono le variabili che leggono la direzione dei joystick rispettivamente di destra e di sinistra e la tramutano nell'opportuno movimento del punto sul video.

La linea 600 controlla che le nuove posizioni non siano già occupate, nel qual caso si va alla routine di fine round, quindi, dopo un rallentamento legato alla velocità scelta (610) e ad un suono (620), si passa alla mossa successiva.

Al termine di ogni round il computer dichiara il vincitore dello stesso, o la parità se entrambi hanno toccato contemporaneamente, ed aggiorna la situazione del punteggio per passare, poi, ad una nuova fase della partita a meno che non si sia raggiunto il numero di round predefinito, nel qual caso si ha la proclamazione del vincitore o di un eventuale salomonico pareggio.

La considerazione che ha spinto il presente lavoro è stata il bisogno di approfondire la conoscenza di alcuni aspetti della programmazione la cui padronanza non è sempre sufficientemente sviluppata a svantaggio dell'uso globale della macchina posseduta. A fianco dei concetti sono stati presentati dei semplici programmi quale esempio pratico delle opportunità offerte dalle tecniche discusse. Ad ognuno, ora, l'approfondimento secondo i canoni personali e lo sviluppo di ulteriori contributi in questo senso. ■

é in edicola

L. 2.000

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana

GRUPPO EDITORIALE JACKSON

2

2 Bacheca
Notizie e notiziole, annunci e iniziative

7 Software
Programmi recensiti per la scuola

33 Corsologo 1/ Il modo dolce
Inizia un corso accelerato di Logo: da non perdere

38 Indagini al calcolatore
Unità didattica per la scuola elementare, dal progetto IRIS del CEDE

12 Limbiate un anno dopo
Alle prese con Logo per il secondo anno consecutivo

8 L'osservazione del comportamento
Ancora sulla valutazione aiutata dal computer: come valutare il comportamento durante una discussione in classe?

45 L'informatica nelle scuole di Roma
L'intervento dell'Assessorato della Provincia per portare l'informatica nelle scuole di Roma.

GRUPPO PER LA VALUTAZIONE DEL SOFTWARE

Un numero sempre maggiore di programmi di software didattico, prodotti da case editrici sia straniere che italiane, si sta rendendo disponibile sul nostro mercato. Nel novero di tale produzione sono compresi programmi dalle caratteristiche spesso assai diverse, e per nulla equivalenti, pur nell'apparente comunanza di materie e di destinazioni. È quindi necessario che gli utenti e in particolare gli insegnanti possano orientarsi nella scelta e nella acquisizione dei programmi più rispondenti alle loro effettive finalità ed esigenze.
Presso la DIDA.EL è stato costituito un gruppo di specialisti, costituito da insegnanti, programmatori e progettisti di software didattico, che utilizzando una guida di valutazione offrono un servizio professionale di consulenza a editori, distributori e privati.
La guida per la valutazione di software didattico è stata sviluppata dalla d.ssa Gianna Dotti Martinengo per la DIDA.EL di Milano che, da alcuni anni, opera specificamente nel campo del software didattico sulla base di esperienze acquisite presso i più avanzati centri di studio statunitensi, allo scopo di disporre di affidabili criteri standard per la propria produzione e di fornire un servizio professionale di consulenza e valutazione.
Tale guida si ispira oltre che ai principi fondamentali della didattica, ai criteri di disegno e progettazione di lezioni con elaboratore riportate dalla letteratura specializzata americana ed europea e alle indicazioni provenienti dalla esperienza DIDA.EL di produzione di più di 300 ore di programmi didattici su elaboratore.

Dida.El, Via Lamarmora 3/A, Milano, tel. 02/ 5459441.

DIDACTICS USA

Dal 7 al 10 novembre il Centro Commerciale Americano ha ospitato presso il padiglione espositivo di Via Gattamelata 5, Milano, la prima edizione della mostra DIDACTICS USA.
La manifestazione, riservata a prodotti americani, offre una panoramica esauriente del più avanzati strumenti didattici oggi disponibili: dai sistemi audiovisivi (hardware e software) ai giochi educativi, dagli impianti per laboratorio e officina ai libri di testo e di didattica.
In concomitanza con la Mostra ed in collaborazione con l'ANCIFAP (Associazione Nazionale Centri IRI - Formazione e Addestramento Professionale), si è tenuto un seminario (*Nuove tecnologie nella didattica: un ponte tra scuola e industria*), così articolato: 8 novembre - La Didattica negli Stati Uniti e in Italia: Esperienze e Confronto.
9 novembre - Sussidi Didattici nell'Apprendimento delle Lingue: English as a Second Language - Approcci ed Esperienze.
Nel corso dei lavori sono intervenuti Mauro Laeng, che ha svolto la relazione introduttiva cordinando i lavori. Nel numero di dicembre relazioneremo ampiamente sull'andamento del seminario e della vivace tavola rotonda svoltas nel pomeriggio. Per ora limitiamo a segnalarvi partecipazioni: Giovan Martorana, Gianpaolo C trone, Antonio Thierry do Gilardi, Maria G Amedeo Monterubbi Vincenzo Giffoni, Giu Medusa, Paolo Pei sare Vaciago, Arrig latta, Matteo Vita.

DIDACTA 85

Dal 25 febbraio al 1 marzo 85 si terrà a Stoccarda il Salone internazionale specializzato per la scuola, l'educazione, l'aggiornamento e la formazione professionale.
Espositori di tutto il mondo porteranno qui la loro produzione, dalla più nota alla più nuova, in fatto di educazione e naturalmente di applicazioni didattiche delle nuove tecnologie.
La fiera, forse la più vasta in Europa, costituirà un momento di reale verifica di quanto il mondo sta facendo per avvicinare l'educazione alla tecnologia. Come dice lo slogan di Didacta: il futuro ha bisogno di conoscenza.

PERSONAL SOFTWARE

Torneo di bridge

— Parte seconda —

Elaborate velocemente le classifiche con il vostro Apple II

di *Vincenzo Delle Cave*

Nella puntata precedente era stato presentato il programma Bridge-Memorizza che curava l'acquisizione degli score e forniva come output due file: File-Torneo e File-Dati, contenenti rispettivamente gli score del torneo e i suoi dati caratteristici (numero turni di giro, numero mani giocate e numero di tavoli). Presentiamo ora, in quest'ultima parte, il programma Bridge-Conti (vedi listato 1) che elabora i dati contenuti nel File-Torneo e fornisce le due classifiche separate (N-S e E-O) e due prospetti rappresentativi di tutte le mani giocate.

Prima di spiegare dettagliatamente le caratteristiche del programma occorre aprire una parentesi per capire come vengono assegnati i punti ad ogni coppia in funzione del risultato conseguito nella singola mano. Converrà, come al solito, fornire un esempio. Consideriamo il seguente score:

5	+420	105
6	+420	103
7	−100	101
8	−50	112
9	−50	110
10	+140	108
11	+170	106
12	−300	104
13	+140	102
1	+170	113
2	+120	111

Il punteggio max. (Top) è calcolato in questo modo: Top = 2(x - 1) dove x rappresenta il numero di turni di gioco, cioè il numero di volte in cui è stata giocata la singola mano.

Il punteggio minimo assegnato (zero) è proprio 0. Nel nostro esempio

Listato 1. *Il programma Bridge-Conti.*

```
10 D$ =  CHR$ (4)
12  DIM PP%(25,120)
13  DIM PA%(41,3): REM     MATRICE
     DEI PUNTEGGI E DELLE COPPIE
     DA INSERIRE
18  DIM MNS%(41,41)
19  DIM MEO%(41,41)
20  DIM A%(41,2): REM   MATRICE D
    EI PUNTEGGI N-S
22  DIM C%(41,2): REM   MATRICE D
    EI PUNTEGGI E-O
25  DIM VHR%(50)
30 ZZ$ = "S"
31 W = 0
35  HOME : INVERSE : PRINT "ATTEN
    DERE PER I CONTI": NORMAL : PRINT
    : PRINT : PRINT
40  GOSUB 8500
50  GOSUB 1977
55  REM
60  GOSUB 7000
70  GOSUB 1697
80  REM
85 X = YY:LL = 1
90  FOR I = 1 TO X
100 PA%(I,1) = MNS%(I,0):PA%(I,2)
     = MNS%(I,N + 1)
110  NEXT I
120  GOSUB 7050
130  PRINT : PRINT D$;"PR#1"
132  PRINT : PRINT "C L A S S I F
     I C A     N - S": PRINT : PRINT
     : PRINT : PRINT
133  GOSUB 140
134  FOR I = 1 TO X
135 PA%(I,1) = MEO%(I,0):PA%(I,2)
     = MEO%(I,N + 1)
136  NEXT I:LL = 1: GOSUB 7050
137  PRINT : PRINT : PRINT "C L A
     S S I F I C A     E - O": PRINT
     : PRINT : PRINT : PRINT
138  GOSUB 140
139  PRINT : PRINT D$;"PR#0": GOTO
     200
140  FOR I = 1 TO X
145  PRINT PA%(I,1);: PRINT  SPC(
     5 -  LEN ( STR$ (PA%(I,1))));
     : PRINT PA%(I,2)
150  NEXT I
160  RETURN
200  PRINT : PRINT D$"RUNBRIDGE-M
     EMORIZZA"
1697  POKE 1273,127
1707  PRINT : PRINT D$;"PR#1"
1717  PRINT  CHR$ (9);"132N"
1727  PRINT  CHR$ (27); CHR$ (20)
1737  PRINT " ": PRINT " ": PRINT
     " ": PRINT " "
1747  REM  STAMPA
1750 S = N
```

sarà Top = 20 e Zero = 0 (non è necessariamente questa la scala usata, a volte è Top = 2 ★ X e Zero = 2).

Gli altri punteggi si ottengono calando ogni volta di due punti il Top; cosicché nel nostro esempio verrà adottata questa scala di valori: 20; 18; 16; 14; 12; 10; 8; 6; 4; 2; 0; che sono usati per assegnare i punti alle undici coppie che hanno giocato la mano. Tutto ciò è molto semplice se le undici coppie ottengono nella stessa mano undici punteggi diversi; questo è un caso piuttosto raro e caratteristico di un torneo poco qualificato. Nel nostro esempio le coppie N-S 5 e 6 hanno ottenuto il punteggio più alto (+ 420); se avessero ottenuto un punteggio diverso avremmo assegnato ad una coppia 20 punti e all'altra 18. Poiché hanno ottenuto entrambe lo stesso punteggio ad ognuna spetteranno 19 punti (le rispettive coppie E-O avversarie otterranno 1 punto ciascuna, e cioé il complemento a 20). Il secondo miglior risultato è stato ottenuto dalle coppie 11 e 1 che prenderanno 15 punti ciascuna (16 e 14 se avessero ottenuto risultati differenti). Seguendo lo stesso criterio per tutte le coppie si finisce per assegnare i seguenti punteggi alle coppie N-S;

N-S	
5	19
6	19
7	2
8	5
9	5
10	11
11	15
12	0
13	11
1	15
2	8

Complementando a 20 i punti ottenuti dalle coppie N-S si ottiene per le coppie E-O.

E-O	
105	1
103	1
101	18
112	15
110	15

```
1757  FOR I = 0 TO S
1760  PRINT  SPC( 3 -  LEN ( STR$
     (MNS%(0,I))))MNS%(0,I);
1777  NEXT I
1778  PRINT " ";"TOT";
1787  PRINT " "
1797  FOR I = 1 TO YY
1807  PRINT " "
1817  FOR J = 0 TO S + 1
1820  IF MNS%(I,J) = 7777 THEN  PRINT
     "  -";: GOTO 1845
1827  PRINT  SPC( 3 -  LEN ( STR$
     (MNS%(I,J))))MNS%(I,J);
1845  IF J = S THEN  PRINT " ";
1847  NEXT J
1848  PRINT " "
1853  NEXT I
1854  FOR I = 1 TO 25
1855  PRINT " "
1857  NEXT
1867  FOR I = 0 TO S
1870  PRINT  SPC( 3 -  LEN ( STR$
     (MEO%(0,I))))MEO%(0,I);
1887  NEXT I
1890  PRINT " ";"TOT";
1897  PRINT " "
1907  FOR I = 1 TO YY
1917  PRINT " "
1927  FOR J = 0 TO S + 1
1930  IF MEO%(I,J) = 7777 THEN  PRINT
     "  -";: GOTO 1955
```

```
1937  PRINT  SPC( 3 -  LEN ( STR$
     (MEO%(I,J))))MEO%(I,J);
1955  IF J = S THEN  PRINT " ";
1957  NEXT J: PRINT " "
1958  NEXT I
1959  FOR I = 1 TO 25
1960  PRINT " "
1961  NEXT
1966  PRINT D$;"PR#0"
1970  RETURN
1977 RIC = X
1980 PUNTMED = X - 1
1981 XX = X + 5
1982 NUM =  INT (YY / 2)
1983 FLAG = YY -  INT (YY / 2) *
     2
1991  FOR I = 1 TO XX
1992 VHR%(I) = 0
1994  NEXT I
1997  REM
2000  FOR P = 0 TO YY
2005 MNS%(P,0) = P
2006 MEO%(P,0) = P + 100
2007  NEXT
2017  FOR P = 0 TO N
2027 MNS%(0,P) = P
2037 MEO%(0,P) = P
2047  NEXT
2077  FOR I = 1 TO YY
2087  FOR J = 1 TO N
2097 MNS%(I,J) = 7777
```

Seguito listato Bridge-Conti.

```
2107 MEO%(I,J) = 7777
2117  NEXT : NEXT
2127  RETURN
2137 W = W + 1
2147  FOR I = 1 TO X
2167 MNS%(A%(I,1),W) = A%(I,2)
2177 MEO%(C%(I,1) - 100,W) = 2 *
     (X - 1) - C%(I,2)
2187  NEXT I
2237  RETURN
7000  FOR NB = 1 TO N
7010  FOR I = 1 TO X
7020 PA%(I,1) = PP%(I,NB * 3 - 2)
7030 PA%(I,2) = PP%(I,NB * 3 - 1)
7040 PA%(I,3) = PP%(I,NB * 3)
7045  NEXT I
7050 F = 0
7060  FOR I = 1 TO X - 1
7070  IF PA%(I,2) >  = PA%(I + 1,
     2) GOTO 7180
7080 T = PA%(I,2)
7090 U = PA%(I,1)
7100 V = PA%(I,3)
7110 PA%(I,1) = PA%(I + 1,1)
7120 PA%(I,3) = PA%(I + 1,3)
7130 PA%(I,2) = PA%(I + 1,2)
7140 PA%(I + 1,2) = T
7150 PA%(I + 1,1) = U
7160 PA%(I + 1,3) = V
7170 F = 1
7180  NEXT I
7190  IF F = 1 THEN 7050
7200  IF LL = 1 THEN LL = 0: RETURN
7300  FOR I = 1 TO X
7310  IF PA%(I,2) =  - 9999 THEN
     MM = MM + 1
7320  NEXT I
7321 TMR = X * 2 - 2 - MM
7322  FOR I = 0 TO X - 2
7323 VHR%(I + 1) = TMR - 2 * I
7324  IF VHR%(I + 1) < 0 THEN VHR
     %(I + 1) = 0
7325  NEXT I
7326  FOR I = X TO XX
7327 VHR%(I) = 0
7328  NEXT I
7380 SUM = VHR%(1)
7390 L = 1
7400  FOR I = 1 TO X - 1
7410  IF PA%(I,2) =  - 9999 THEN
     7585
7420  IF PA%(I,2) <  > PA%(I + 1,
     2) GOTO 7460
7430 SUM = SUM + VHR%(I + 1)
7440 L = L + 1
7450  GOTO 7550
7460  FOR J = I TO I - L + 1 STEP
     - 1
7470 A%(J,2) = SUM / L
7480 A%(J,1) = PA%(J,1)
7490 C%(J,2) = SUM / L
7500 C%(J,1) = PA%(J,3)
7510  NEXT J
7520 L = 1
7530 SUM = VHR%(I + 1)
7540  IF I = X GOTO 7630
7550  NEXT I
7560  IF PA%(X,2) =  - 9999 GOTO
     7585
7570 I = X
7580  GOTO 7460
7585  FOR AA = I TO I + MM - 1
7590 A%(AA,2) = PUNTMED
7600 A%(AA,1) = PA%(AA,1)
7610 C%(AA,2) = PUNTMED
7620 C%(AA,1) = PA%(AA,3)
7625  NEXT AA
7628 MM = 0
7630  GOSUB 2137: REM    ASSEGNAZ
     IONE PUNTEGGI
7640  REM  SOMMA PUNTEGGI NS
7755  PRINT "*";
7760  NEXT NB
7770  REM  ASSEGNA TOTALI
7780  FOR I = 1 TO YY
7781  FOR J = 1 TO N
7782  IF MNS%(I,J) = 7777 THEN KR
     = 0: GOTO 7784
7783 KR = MNS%(I,J)
7784  IF MEO%(I,J) = 7777 THEN KZ
     = 0: GOTO 7787
7785 KZ = MEO%(I,J)
7787 SNS = SNS + KR
7788 SEO = SEO + KZ
7789  NEXT J
7790 MNS%(I,N + 1) = SNS
7800 MEO%(I,N + 1) = SEO
7802 SNS = 0:SEO = 0
7810  NEXT I
7820  RETURN
8500  REM
8501  PRINT : PRINT D$"OPEN DATI"

8502  PRINT D$"READ DATI"
8504  INPUT X: INPUT N: INPUT YY
8505  PRINT D$"CLOSE DATI"
8510  PRINT : PRINT D$;"OPEN TORN
     EO"
8520  PRINT D$;"READ TORNEO"
8530  FOR J = 1 TO N * 3
8540  FOR I = 1 TO X
8550  INPUT PP%(I,J)
8560  NEXT I: NEXT J
8570  PRINT D$;"CLOSE TORNEO"
8580  RETURN
```

108	9
106	5
104	20
102	9
113	5
111	12

Uno score può a volte presentare delle complicazioni dovute al fatto che il direttore del torneo, per svariati motivi, assegna un punteggio medio alle due coppie N-S e E-O che giocano la mano. In tal caso per calcolare la scala dei punteggi si deve calare di 1 il top per ogni mano media assegnata e di conseguenza aumentare di 1 lo zero. Pertanto, nel nostro esempio, la scala di punteggi da assegnare per uno score che preveda una mano media è la seguente: 19; 17; 15; 13; 11; 9; 7; 5; 3; 1; mentre 10 è il punteggio da assegnare alla mano media (MM).
Espresso in formule:
Top = 2★(X-1) - N. Mani Medie
Zero = 0 + N. Mani Medie
MM = (Top + Zero)/2
Notare che la somma dei punteggi che vengono attribuiti a tutte le coppie è costante, indipendente dallo score ed è uguale a: MM ★ X, cioè punteggio medio per numero di volte in cui è stata giocata la mano. Nel nostro caso sarà 110.

REMarks

8500-8580 - Per prima cosa viene letta dal File-Torneo la matrice PP% rappresentativa degli score e caricati, dal File-Dati, il numero di turni, numero di mani e numero di tavoli del torneo.
1977-2237 - Inizializzazione delle matrici MNS% e MEO% rappresentative dei prospetti finali (prima colonna inizializzata al numero della coppia e prima riga al numero della mano. Le matrici sono riempite con dei valori convenzionali (7777) che verranno poi convertiti in "-" all'atto della stampa se la coppia non ha giocato quella mano).
7000-7820 - È il nucleo del programma. Viene calcolato il punteggio di ciascuna coppia dello score. Inizialmente è previsto un ordinamento decrescente dei punteggi (linee 7010-7190), quindi il calcolo delle eventuali mani medie (7300-7320) e la routine di calcolo dello score (7321-7628); infine è prevista l'assegnazione del punteggio ottenuto da ogni coppia al prospetto finale (2137-2237). Per ogni coppia viene calcolata la somma dei punteggi ottenuti e memorizzata nell'ultima colonna

0	1	2	3	4	5	6	7	8	TOT	
1	9	10	15	10	20	2	5	5	76	
2	-	10	10	14	18	20	8	3	83	
3	-	-	13	14	14	18	10	1	70	
4	-	-	-	14	16	14	20	19	83	
5	-	-	-	-	7	8	12	17	44	
6	-	-	-	-	-	2	15	7	24	
7	-	-	-	-	-	-	18	14	32	
8	-	-	-	-	-	-	-	10	10	
9	-	-	-	-	-	-	-	-	0	
10	-	-	-	-	-	-	-	-	0	
11	18	-	-	-	-	-	-	-	18	
12	14	5	-	-	-	-	-	-	19	
13	9	5	7	-	-	-	-	-	21	
14	20	18	10	2	-	-	-	-	50	
15	3	9	18	4	12	-	-	-	46	
16	0	16	18	18	10	8	-	-	70	
17	3	13	5	6	0	14	1	-	42	
18	14	13	10	9	2	14	5	14	81	
19	14	2	2	9	4	2	15	10	58	
20	6	9	2	10	7	8	1	10	53	

0	1	2	3	4	5	6	7	8	TOT	
101	11	13	9	-	6	-	0	-	39	
102	-	8	12	4	-	6	-	2	32	
103	6	-	6	16	4	-	8	-	40	
104	-	13	-	4	8	12	-	12	49	
105	11	-	9	-	13	12	5	-	50	
106	-	0	-	14	-	18	19	5	56	
107	0	-	1	-	10	-	2	5	18	
108	-	9	-	0	-	6	-	9	24	
109	17	-	1	-	20	-	15	-	53	
110	-	2	-	12	-	6	-	9	29	
111	20	-	14	-	18	-	5	-	57	
112	-	5	-	9	-	18	-	9	41	
113	17	-	9	-	16	-	19	-	61	
114	-	5	-	9	-	12	-	14	40	
115	6	-	17	-	13	-	15	-	51	
116	-	16	-	8	-	18	-	16	58	
117	6	-	17	-	0	-	12	-	35	
118	-	9	-	8	-	0	-	18	35	
119	14	-	4	-	2	-	10	-	30	
120	2	8	-	4	-	2	-	0	16	

Figura 1. *I prospetti finali di un torneo a 8 mani e 20 coppie.*

Fiera del Levante, Comufficio e Tecnopolis hanno creato a Bari

TECNORAMA/UFFICIO

PERCHÉ É PIÚ REDDITIZIO PORTARE I PRODOTTI SUL MERCATO CHE IL MERCATO SUI PRODOTTI

Il mercato del Mezzogiorno: un mercato geograficamente lontano dagli importanti Saloni internazionali, ma un mercato fecondo perchè sulla via di un rapido rinnovamento.
Per questo Fiera del Levante, COMUFFICIO e Tecnopolis - sotto il patrocinio dello SMAU e delle Università del Mezzogiorno - oggi presentano TECNORAMA/UFFICIO: la nuova rassegna di macchine, attrezzature e arredamenti per l'ufficio, che si terrà a Bari dal 24 al 28 gennaio 1985.
Una rassegna di 35.000 metri quadri, ricca di iniziative promozionali, che stabilirà contatti diretti fra produttori e migliaia di aziende agricole, industriali e commerciali operanti nell'intero territorio meridionale e desiderose di innovare e di rinnovarsi.
Una rassegna che creerà nuovi legami fra operatori del settore e migliaia di studi professionali ed amministrativi.
Una rassegna che si rivolgerà anche e particolarmente al mondo dei giovani.
Chi decide di partecipare a TECNORAMA/UFFICIO come espositore decide anche di cogliere una occasione importante per un importante sviluppo dei propri affari in un'area geografica quanto mai importante, che si estende, grazie all'esperienza della Fiera del Levante, fino ai Paesi del Mediterraneo.
Perchè è più redditizio portare i prodotti su un mercato in attesa che non viceversa.

TECNORAMA UFFICIO

SALONE DELLA TELEMATICA E DELLE COMUNICAZIONI NELL'ORGANIZZAZIONE AZIENDALE: MACCHINE ED ARREDAMENTI PER L'UFFICIO

E.A. FIERA DEL LEVANTE
LUNGOMARE STARITA 70123 BARI
TELEFONO (080) 206111
TELEX 810101 FIDLEV I

Syncronia

CLASSIFICA N-S		CLASSIFICA E-O	
2	83	113	61
4	83	116	58
18	81	111	57
1	76	106	56
3	70	109	53
16	70	115	51
19	58	105	50
20	53	104	49
14	50	112	41
15	46	103	40
5	44	114	40
17	42	101	39
7	32	117	35
6	24	118	35
13	21	102	32
12	19	119	30
11	18	110	29
8	10	108	24
9	0	107	18
10	0	120	16

Figura 2. *Le classifiche N-S e E-O del torneo di figura 1.*

del prospetto finale (7770-7810).

1697-1970 - Stampa su 132 colonne dei prospetti finali. Le linee 1677-1727 prevedono l'inizializzazione per la stampante Centronics 739 e vanno opportunamente modificate per le altre stampanti.

85-200 - Stampa della classifica finale delle coppie N-S e E-O: terminata questa operazione il comando ritorna al programma Bridge-Memorizza.

Conclusioni

Prima di terminare accenniamo brevemente alla velocità di esecuzione del programma, una volta terminata la fase di introduzione dei dati (che, prevedendo la possibilità di inserimento parziale dei risultati, può terminare entro dieci minuti dalla fine del torneo).

Per il calcolo di uno score contenente 20 risultati è necessario un tempo di elaborazione di circa 7 secondi, per cui per un torneo di 40 mani si ottiene la classifica finale dopo circa quattro minuti; tale tempo si riduce di oltre la metà se il programma viene compilato (ad esempio usando il Task lo stesso score viene elaborato in meno di 3 secondi). Come termine di paragone si consideri che una persona esperta eseguendo i conti a mano da sola impiegherebbe mediamente circa 3 ore per ottenere i risultati completi.

RIVISTE JACKSON.
LA VOCE
PIÙ AUTOREVOLE
NEL CAMPO
DELL'ELETTRONICA
E DELL'INFORMATICA.

l'Elettronica
PERSONAL SOFTWARE
AUTOMAZIONE
strumenti MUSICALI
INFORMATICA
elektor
Bit
VIDEO GIOCHI
elettronica OGGI
Telecomunicazioni

GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

Archivio per Spectrum

Un utile strumento per memorizzare i vostri dati

di *Natale Carloni*

Il titolo dice già tutto, o quasi, della sua funzione, con la sola aggiunta che esso è rivolto ad una particolare categoria, anche se certamente non esigua, di utenti di questo tipo di software.
La molla, anzi l'assillo, che mi ha spinto a stendere il programma è stata la necessità che spesso mi si presenta di reperire tra i lavori che possiedo quelli che di volta in volta mi servono (lavori, articoli, reprint, libri, ecc.).
Ora, tutti coloro che per la loro attività o semplicemente per interesse personale, sono abbonati a riviste scientifiche, raccolgono fotocopie di lavori e materiale bibliografico vario, conoscono molto bene questo problema. Il materiale cartaceo aumenta costantemente (in qualche caso addirittura geometricamente), e se fin quando si dispone solo di poche centinaia di titoli è ancora agevole archiviare e poi ripescare quello che serve, ciò diviene sempre più tormentoso mano a mano che i lavori si accumulano. Quanti non si sono mai chiesti di fronte a un nuovo lavoro: "Questo lo metto nella sezione A o nella sezione B? Lo metto nella A con un riferimento nella B? oppure viceversa? Apro una nuova sezione? ecc.".
Fino a quando, presi dallo sconforto, ci si rassegna a conservare solo ciò che al momento sembra rilevante (falsa soluzione!: dopo 2 giorni si torna a raccogliere tutto) oppure ci si affida alle capacità mnemoniche del proprio cervello (che però, si rifiuta di ricordare una simile mole di dati) e si verifica, praticamente, il caos.

Con questo programma ho risolto gran parte dei miei problemi e riesco a consultare tutti i miei lavori (circa 3.000 titoli nell'ambito delle scienze mediche) e a reperire quelli che mi interessano.

Il programma è costituito da un primo blocco introduttivo che effettua una semplice inizializzazione (BORDER, PAPER, INK, CLEAR 32767, ecc.) e presenta la lista delle opzioni, e di altri sette brevissimi blocchi ciascuno dei quali espleta un'opzione.
Ecco l'elenco che viene presentato all'inizio:

ARCHIVIO

1 Immissione
2 Ricerca per autore
3 Ricerca per titolo
4 Ricerca per chiave
5 Ricerca per collocazione
6 Correzioni

Premendo il numero corrispondente a una delle opzioni si dispone il computer a caricare, fra i blocchi che seguono, solo quello con il programma relativo all'opzione prescelta.
Se si ha l'accortezza di lasciare un intervallo di circa un minuto tra l'introduzione e il primo dei blocchi successivi non è neppure necessario fermare il registratore durante la scelta dell'opzione. Premendo il 6 viene visualizzato quest'altro breve elenco preliminare:

CORREZIONI

1 Lavori in posizione nota
2 Lavori in posizione sconosciuta

I lavori vengono memorizzati in matrici che contengono 100 record ognuna - DIM B$ (100, 81). La suddivisione del record è effettuata in figura 1.
Durante l'esecuzione del programma le matrici vengono caricate (e/o salvate a secondo dell'opzione) in blocchi autonomi rispetto al programma - LOAD/SAVE "arc" DATA B$ ().

1) Immissione

All'inizio viene richiesto se si desidera immettere dati in un blocco nuovo, oppure in uno già iniziato. Nel secondo caso il computer chiede di caricare il blocco già iniziato, individua l'ultimo lavoro presente nel blocco e si dispone per accogliere i dati del nuovo lavoro nel primo record libero. L'immissione avviene attraverso una maschera nella quale il cursore si colloca di volta in volta a fianco del nome del dato richiesto. Il numero in chiaro in alto a sinistra (n) è il numero d'ordine del record all'interno della matrice (da 1 a 100). Premendo ENTER dopo ogni dato il cursore si sposta al dato successivo e, dopo l'ultimo dato del record, viene visualizzata la maschera vuota

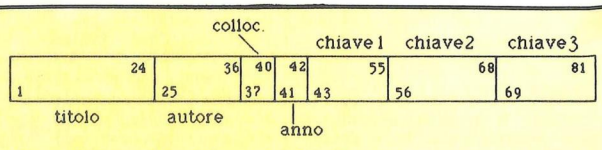


Figura 1. *Il tracciato del record dell'archivio.*

Listato 1. *Il listato del programma Archivio suddiviso in blocchi.*

ARCHIVIO

```
  10 OVER 1
  20 LET a=-1: LET b=177: LET c=
256: LET d=176
  30 FOR j=0 TO 36
  40 LET a=a+2: LET b=b-4: LET c
=c-2: LET d=d-2
  50 PLOT a,a: DRAW 0,b: PLOT c,
d: DRAW 0,-b
  60 NEXT j
  70 RETURN
  80 PRINT AT 9,12;"ARCHIVIO";AT
 11,15;"by";AT 12,11;"N. Carioni
"
  90 RETURN
 100 FOR a=0 TO 254 STEP 2: PLOT
 a,111: DRAW 0,-47: PLOT 255-a,6
4: DRAW 0,47
 110 NEXT a
 120 RETURN
 130 CLEAR 32767: BORDER 6: PAPE
R 6: INK 0: CLS : GO SUB 20
 140 GO SUB 80: GO SUB 10: GO SU
B 20: GO SUB 10: GO SUB 80
 150 FOR n=8 TO 13
 160 PRINT AT n,0;"
                  ": NEXT n
 170 PRINT BRIGHT 1;AT 9,0;"
  SCEGLIERE L'OPZIONE"''"     SE
NZA FERMARE IL NASTRO"
 180 PAUSE 40: GO SUB 100: PAUSE
 80: GO SUB 100: PAUSE 40: OVER
0
 190 CLS : PRINT AT 1,12;" ARCHI
VIO "''TAB 5;"1  INTROD. NUOVI
TITOLI"''TAB 5;"2  RICERCA PER A
UTORE"''TAB 5;"3  RICERCA PER TI
TOLO"''TAB 5;"4  RICERCA PER CHI
AVE"''TAB 5;"5  RICERCA PER COLL
OC."''TAB 5;"6  CORREZIONE"
 200 INK 6: LET a=CODE INKEY$
 210 IF a<49 OR a>54 THEN GO TO
200
 220 IF a=49 THEN LOAD "archivia
"
 230 IF a=50 THEN LOAD "autore"
 240 IF a=51 THEN LOAD "titolo"
 250 IF a=52 THEN LOAD "chiave"
 260 IF a=53 THEN LOAD "colloc."
 270 CLS : PRINT INK 0;"       NON
```

del primo record libero successivo. Se durante l'immissione di un record ci si rende conto di un errore commesso nel dato che si sta introducendo, lo si può facilmente correggere con l'uso del DELETE e delle frecce a sinistra e a destra. Se l'errore è in un dato già immesso, si dia uno STOP premendo Caps SHIFT e 6; con un successivo CONTINUE l'immissione riprende dall'inizio dello stesso record.

Terminata l'immissione del record n. 100, il blocco ultimato viene registrato (è sufficiente far partire il registratore e premere un tasto alla comparsa del consueto Start Tape). Comunque è possibile interrompere l'immissione anche prima di essere arrivati al record 100 (per esempio se ci si stanca o se i lavori da registrare sono terminati). Per questo è sufficiente introdurre un record in "bianco", vale a dire non immettere dati, ma premere solo ENTER: al termine di questo record il computer non presenta la maschera del record successivo ed inizia il salvataggio di questo blocco-dati parziale. Al termine del salvataggio viene richiesto "Continui?"; se sì la procedura ricomincia, se no il programma si arresta.

Attenzione: il programma considera vuoto un record se è vuota la sua posizione iniziale (che è anche la prima del titolo): perciò nell'archiviazione di un lavoro si può tralasciare di immettere alcuni dati (perché, per esempio, non si conosce l'anno di pubblicazione oppure l'autore, oppure sono sufficienti meno parole-chiave di quelle predisposte), ma non il titolo: se non si conosce il titolo di un lavoro che si intende registrare è comunque necessario mettere qualcosa nella prima posizione (per esempio un asterisco, un punto, un trattino).

Per quanto riguarda l'ampiezza dei singoli campi, da quanto detto a proposito della suddivisione del record si ottiene:

Collocazione:	4 caratteri
Autore:	12 caratteri
Titolo:	24 caratteri
Anno:	2 caratteri
Chiavi:	13 caratteri per ciascuna

2) Ricerca per autore

Questa opzione richiede il nome di un autore e individua nei blocchi-dati tutti i suoi lavori. L'operazione avviene confrontando il nome immesso per la ricerca con i nomi presenti nei diversi record. È superfluo ricordare che la presenza anche di un solo carattere diverso fa sì che quel lavoro non venga individuato. Per ridurre questo inconveniente all'inizio viene richiesto "CUT 8/12?", che significa se si desidera ricercare un autore sulla base solo delle prime 8 posizioni anziché tutte e 12 (cut = taglio). In questo modo, anche se generalmente non è necessario ricorrere ad abbreviazioni per l'autore, è possibile ritrovare i lavori di un autore per esempio dal cognome troppo lungo, anche se non ci si

ricorda come lo si è abbbreviato, oppure se ci si ricorda solo il cognome e non il nome. Un metodo standard per scrivere l'autore è quello di riportare per primo il cognome (con iniziale maiuscola) e di seguito, ma non prima dell'ottava posizione (così da poter essere eventualmente escluso dal CUT), l'iniziale maiuscola del nome, seguita a sua volta da "al" (che stà per "alii") se vi sono coautori. Alcuni esempi:

- Brody J;
- Elizan T;
- Feldman Ral.

I lavori dell'autore richiesto vengono visualizzati con il formato della figura 2.

Quando tutti i record di un blocco sono stati esaminati si ha la scritta "Fai partire il registratore", per caricare il blocco successivo e poi, a operazione ultimata, la ricerca riprende automaticamente.

Nell'operazione di salvataggio che conclude la fase di immissione è stata introdotta volutamente una pausa prolungata tra la richiesta di accendere il registratore e l'inizio della registrazione vera e propria. Conseguentemente sul nastro tra un blocco-dati e il successivo vi è un sufficiente intervallo perché il computer effettui la ricerca completa e si disponga nuovamente per il caricamento, senza che nel frattempo sia necessario spegnere il registratore. Sarà sufficiente fermare il nastro solo nel caso che i titoli trovati abbiano riempito lo schermo (nel qual caso viene richiesto uno Scroll).

3) Ricerca per titolo

Opera esattamente come l'opzione precedente, con la differenza che il titolo può essere ricercato sulla base di tutti i 24 caratteri, oppure solo dei primi 16 (rispondendo "si" alla domanda "CUT 16/24?").

4) Ricerca per collocazione

Anche questo programma opera in modo analogo a quanto visto al punto 2. Però in questo caso, visto che i singoli caratteri di questo campo possono avere i significati più svariati (a secondo del tipo di codificazione che si intende adottare), vi è la possibilità di effettuare la ricerca in base a 1, 2, 3 oppure tutti e 4 i caratteri della collocazione.

Per questo all'inizio vi è un'apposita richiesta.

5) Ricerca per chiave

Per l'individuazione di un lavoro in base a una parola-chiave (la keyword degli anglosassoni) è sufficiente che una delle sue tre chiavi corrisponda con quella immessa in fase di ricerca. Se non si desidera confrontare tutta la parola, si può richiedere un CUT 8/13, limitando così il confronto alle prime 8 lettere. Sfruttando questa particolarità è possibile rendere la ricerca per parole chiave più o meno specifica. Facciamo un esempio: se abbiamo tre lavori che parlano di ulcera gastrica, potremo usare per ciascuno di essi la parola chiave "ulcera gastr.", e in fase di ricerca reimmettendo la stessa parola (o anche la stessa ma abbreviata con un CUT), ritroveremo tutti e tre i lavori. Tuttavia se ci si rende conto che questi lavori trattano il primo degli aspetti chirurgici, il secondo di quelli farmacologici e il terzo di quelli dietetici, ad esempio, dell'ulcera gastrica, si potrà modificare le parole chiave in questo modo: "ulcera g★chir", "ulcera g★farm" e "ulcera g★diet". Ancora si potrà trovare tutti e tre i lavori usando il CUT e introducendo per la ricerca "ulcera g"; ma qualora ci interessasse un argomento particolare, introducendo il termine completo, dei tre si troverebbe solo il lavoro più attinente.

6) Correzioni

6a) - Lavori in posizione nota.

Si utilizza quando si deve correggere un lavoro del quale si conosce il numero d'ordine all'interno del blocco.

Prima viene richiesto di caricare il blocco, e poi di immettere il numero d'ordine del lavoro da correggere.

Si possono effettuare anche più correzioni nello stesso record. Il puntino in basso a destra sopra la riga dell'Input, corrisponde all'ultima posizione che viene accettata nella sottostringa che corrisponde al titolo; quello che viene scritto più a destra del puntino non viene accettato.

Al termine si preme lo 0, dopodiché il computer chiede se ci sono altre correzioni nello stesso blocco, e se sì, quale numero. In caso contrario si passa al salvataggio del blocco corretto al posto di quello sbagliato.

6b) - Lavori in posizione sconosciuta.

Si utilizza quando, pur sapendo che

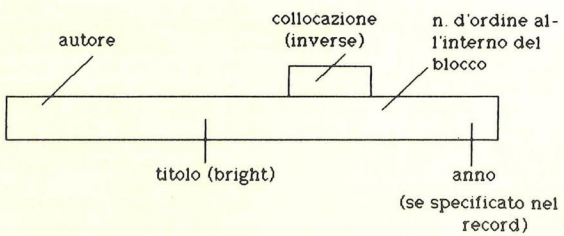


Figura 2. *Il formato della visualizzazione dei record ricercati per autore.*

Seguito listato Archivio.

```
 FERMARE  IL NASTRO        """"""
        CORREZIONI """"" 1  LAV
ORI IN POSIZIONE NOTA"""" 2  LAV
ORI DA RICERCARE"
 280 LET a=CODE INKEY$
 290 IF a<>49 AND a<>50 THEN GO
TO 280
 300 IF a=49 THEN LOAD "unknown"
 310 LOAD "lookfor"
```

CHIAVE

```
  10 INK NOT PI: GO SUB CODE "x"
: INPUT "CUT 8/13 ? ";p$: DIM a$
(VAL "8"+(VAL "5" AND p$<>"s")):
 INPUT "Chiave ____________";"____
" AND p$<>"s";AT SGN PI,VAL "8";
 LINE a$
  20 BEEP .3,CODE "$": PRINT AT
INT PI,VAL "4";"      CASSETTA BLO
CCHI    """      POSIZIONATA ALL'
INIZIO "
  30 PRINT '"  Fai partire  il r
egistratore": INK VAL "6": LOAD
"arc" DATA b$(): CLS : INK NOT P
I: GO SUB CODE "x"
  40 PRINT "Sto cercando i lavor
i con chiave"; BRIGHT SGN PI;a$'
  50 FOR n=SGN PI TO CODE "d"
  60 FOR r=CODE "+" TO CODE "Q"
STEP VAL "13"
  70 IF b$(n,r TO r+LEN a$-SGN P
I)=a$ THEN GO SUB CODE "n"
  80 NEXT r
  90 NEXT n
 100 GO TO VAL "30"
 110 PRINT ''b$(n,VAL "25" TO CO
DE "$"), INVERSE SGN PI;b$(n,COD
E "%" TO CODE "(");: BEEP .1,COD
E "$": PRINT n' BRIGHT SGN PI;b$
(n, TO VAL "24");" 19"+b$(n,CODE
 ")" TO CODE "*") AND b$(n,CODE
")")<>" ": RETURN
 120 BEEP .3,CODE " ": CLS : PRI
NT AT INT PI,VAL "12";" STOP ":
PAUSE CODE "d": CLS : RETURN
```

AUTORE

```
  10 INK NOT PI: GO SUB CODE "Z"
: INPUT "CUT 8/12 ? ";p$: DIM a$
(VAL "8"+(VAL "4" AND p$<>"s"));
 INPUT "Autore ____________";"____
 AND p$<>"s";AT SGN PI,VAL "8";
LINE a$
  20 BEEP .3,CODE "$": PRINT AT
INT PI,VAL "4";"      CASSETTA BLO
CCHI    """      POSIZIONATA ALL'
INIZIO "
  30 PRINT '"  Fai partire  il r
egistratore": INK VAL "6": LOAD
"arc" DATA b$(): CLS : INK NOT P
I: GO SUB CODE "Z"
  40 PRINT "Sto cercando i lavor
i di"' BRIGHT SGN PI;a$'
  50 FOR n=SGN PI TO CODE "d"
  60 IF b$(n,VAL "25" TO VAL "25
"+LEN a$-SGN PI)=a$ THEN GO SUB
CODE "P"
  65 NEXT n
  70 GO TO VAL "30"
  80 PRINT ''b$(n,VAL "25" TO CO
DE "$"), INVERSE SGN PI;b$(n,COD
E "%" TO CODE "(");: BEEP .1,COD
E "$": PRINT n' BRIGHT SGN PI;b$
(n, TO VAL "24");" 19"+b$(n,CODE
 ")" TO CODE "*") AND b$(n,CODE
")")<>" ": RETURN
  90 BEEP .3,CODE " ": CLS : PRI
NT AT INT PI,VAL "12";" STOP ":
PAUSE CODE "d": CLS : RETURN
```

TITOLO

```
  10 INK NOT PI: GO SUB CODE "Z"
: INPUT "CUT 16/24 ? ";p$: DIM a
$(VAL "16"+(VAL "8" AND p$<>"s")
): INPUT "Titolo ________________
";"__________" AND p$<>"s";AT SGN
 PI,VAL "7"; LINE a$
  20 BEEP .3,CODE " ": PRINT AT
INT PI,VAL "4";"      CASSETTA BLO
CCHI    """      POSIZIONATA ALL'
INIZIO "
  30 PRINT '" 'Fai partire  il r
egistratore": INK VAL "6": LOAD
"arc" DATA b$(): CLS : INK NOT P
I: GO SUB CODE "Z"
  40 PRINT "Sto cercando il tito
lo"' BRIGHT SGN PI;a$'
  50 FOR n=SGN PI TO CODE "d"
  60 IF b$(n, TO LEN a$)=a$ THEN
 GO SUB CODE "P"
  65 NEXT n
  70 GO TO VAL "30"
  80 PRINT ''b$(n,VAL "25" TO CO
DE "$"), INVERSE SGN PI;b$(n,COD
E "%" TO CODE "(");: BEEP .1,COD
E "$": PRINT n' BRIGHT SGN PI;b$
(n, TO VAL "24");" 19"+b$(n,CODE
 ")" TO CODE "*") AND b$(n,CODE
")")<>" ": RETURN
  90 BEEP .3,CODE " ": CLS : PRI
NT AT INT PI,VAL "12";" STOP ":
PAUSE CODE "d": CLS : RETURN
```

COLLOCAZIONE

```
  10 INK NOT PI: CLS : PRINT 'TA
B VAL "7";"Seleziona il CUT"'''
'TAB VAL "10";"""- - - -"""'TAB
 VAL "11";: FOR n=SGN PI TO VAL
"4": PRINT FLASH NOT c+SGN PI<n;
n; FLASH NOT PI;" ";: NEXT n: RE
TURN
  20 LET c=-SGN PI: GO SUB VAL "
10"
  30 LET c=CODE INKEY$-CODE "1"
  40 IF c<NOT PI OR c>INT PI THE
N GO TO VAL "30"
  50 GO SUB VAL "10": DIM a$(c+S
GN PI): INPUT "Colloc. ";a$
  60 BEEP .3,CODE "$": CLS : PRI
NT AT INT PI,VAL "4";" INIZIO CA
SSETTA BLOCCHI "
  70 PRINT '"  Fai partire  il r
egistratore": INK VAL "6": LOAD
"arc" DATA b$(): CLS : INK NOT P
I: GO SUB CODE "■"
  80 PRINT "Sto cercando i lavor
i con collo-cazione  "; BRIGHT S
GN PI;a$'
  90 FOR n=SGN PI TO CODE "d"
 100 IF b$(n,CODE "%" TO CODE "%
"+c)=a$ THEN GO SUB CODE "■"
 110 NEXT n
 120 GO TO CODE "F"
 130 PRINT ''b$(n,VAL "25" TO CO
DE "$"), INVERSE SGN PI;b$(n,COD
E "%" TO CODE "(");: BEEP .1,COD
E "$": PRINT n' BRIGHT SGN PI;b$
(n, TO VAL "24");" 19"+b$(n,CODE
 ")" TO CODE "*") AND b$(n,CODE
")")<>" ": RETURN
 140 CLS : PRINT AT INT PI,VAL "
12";" STOP ": BEEP .3,CODE " ":
PAUSE CODE "d": CLS : RETURN
```

ARCHIVIA

```
  10 CLS : INK NOT PI: DIM b$(CO
DE "d",CODE "Q"): INPUT "Blocco
nuovo? ";s$: IF CODE s$=CODE "s"
 THEN GO TO VAL "30"
  20 CLS : PRINT "Carica il bloc
co": INK VAL "6": LOAD "arc" DAT
```

Archivio per Spectrum

Seguito listato Archivio.

```
A b$(): INK NOT PI: CLS
  30 FOR a=SGN PI TO CODE "d": I
F CODE b$(a)=CODE " " THEN LET s
$="                    ": INPUT BRIGHT S
GN PI;AT NOT PI,NOT PI;(a)' BRIG
HT NOT PI;"Coll. ____ Autore ";
(s$)' ""Titolo ";(s$);(s$)' ""19__
 Chiavi:          ";(s$)'(s$);"_
  ";(s$);AT SGN PI,VAL "6";"_LIN
E b$(a,CODE "%" TO CODE "(");AT
SGN PI,VAL "19"; LINE b$(a,VAL "
25" TO CODE "$");AT VAL "3",VAL
"7"; LINE b$(a, TO VAL "24");AT
VAL "5",VAL "2"; LINE b$(a,CODE
")" TO CODE "*";AT VAL "5",VAL
"18"; LINE b$(a,CODE "+" TO CODE
"7");AT VAL "5",NOT PI; LINE b$
(a,CODE "8" TO CODE "D");AT VAL
"6",VAL "18"; LINE b$(a,CODE "E"
 TO CODE "0")
  40 IF CODE b$(a)<>CODE " " THE
N NEXT a
  50 PRINT "Salva questo blocco"
: PAUSE 500
  60 CLS : SAVE "arc" DATA b$():
 INPUT "Continui? ";s$: IF CODE
s$=CODE "s" THEN GO TO VAL "10"
  70 STOP
```

```
UNKNOWN

   1 DATA "..Coll.",CODE "%",COD
E "("
   2 DATA ".Autore",VAL "25",COD
E "$"
   3 DATA ".Titolo",PI=PI,VAL "2
4"
   4 DATA "...Anno",CODE ")",COD
E "*"
   5 DATA "...Ch.1",CODE "+",COD
E "7"
   6 DATA "...Ch.2",CODE "8",COD
E "D"
   7 DATA "...Ch.3",CODE "E",COD
E "0"
  10 INK NOT PI: CLEAR : PRINT "
Carica il blocco": INK VAL "6":
LOAD "arc" DATA b$(): INK NOT PI
: CLS
  30 CLS : INPUT "Numero? ";a
  30 GO SUB CODE "n"
  40 LET g=CODE INKEY$-CODE "0":
 IF g<NOT PI OR g>VAL "7" THEN G
O TO CODE "("
  50 IF g=NOT PI THEN GO TO CODE
 "P"
  60 GO SUB CODE "x"
  70 RESTORE g: READ c$: READ d:
 READ e: INPUT b$(a,d TO e): GO
TO VAL "30"
  80 INPUT "Altre corr.? ";a$: I
F CODE a$=CODE "s" THEN GO TO VA
L "20"
  90 CLS : SAVE "arc" DATA b$():
 INPUT "Altri blocchi? ";a$: IF
CODE a$<>CODE "s" THEN STOP
 100 CLS : GO TO VAL "10"
 110 LET g=NOT PI
```

vi è un lavoro da correggere, non si sà in quale posizione si trova nel blocco.
Sullo schermo viene visualizzato il primo lavoro, in modo analogo a quanto visto al punto 6a, con l'aggiunta di:

8 NEXT

Cioè premendo 8 si passa alla visualizzazione del record sucessivo, e si può così proseguire fino a quando non viene trovato il lavoro da modificare. Se non si desidera proseguire, con lo 0 si registra il blocco dopo avere effettuato la correzione.
Questa opzione è anche molto utile per visualizzare tutti i lavori contenuti in un blocco (per controllo, verifica, ecc.).

Se è necessario cancellare un cospicuo numero di lavori in un blocco, per quanto già visto al punto 1, è sufficiente cancellare il titolo di questi. Ripassando in seguito a questo blocco in fase di immissione, tutti questi record verranno considerati vuoti, pertanto completamente riscrivibili.

Osservazioni sul listato

Il listato, non è molto elegante, ma è stato necessario utilizzare tutti gli espedienti utili per ridurre al minimo l'area di memoria occupata dal programma con le sue variabili. Così il numero delle linee è in genere il minore possibile e inoltre sono praticamente stati eliminati tutti i numeri come tali, che occupano troppi byte, sostituiti dai CODE, dai VAL e altri espedienti. Tutto ciò è necessario per consentire di lavorare con matrici così ampie (più di 8.100 byte l'una).
I singoli blocchi hanno l'autostart; nell'ordine in cui si trovano sul nastro, questi i loro titoli (listato 1):

Archivio	line 130
Chiave	line 10
Autore	line 10
Titolo	line 10
Colloc.	line 20
Archivia	line 10
Unknown	line 10
Lookfor	line 10

I tratti evidenziati alle righe 160, 190 e 270 del programma Archivio si ottengono premendo al loro inizio Caps SHIFT + Symbol SHIFT e successivamente il 9; al loro termine Caps SHIFT + Symbol SHIFT e successivamente l'8. Inoltre, alla riga 270, per dare il lampeggio alla scrittura non fermare il nastro, in aggiunta vanno premuti, all'inizio Caps SHIFT + Symbol SHIFT seguiti da Caps SHIFT + 9, alla fine Caps SHIFT + Symbol SHIFT seguita da Caps SHIFT + 8.
Per quanto riguarda il programma Chiave occorre adottare la seguente procedura alla riga 20: all'inizio dei due tratti evidenziati: Caps SHIFT + Symbol SHIFT e poi 9; al termine: Caps SHIFT + Symbol SHIFT e poi 8. Alla riga 120 per ottenere il lampeggio: all'inizio Caps SHIFT + Symbol SHIFT e poi 9, Caps SHIFT + Symbol SHIFT e poi Caps SHIFT + 9; al termine Caps SHIFT + Symbol SHIFT e poi 8, Caps SHIFT + Symbol SHIFT e poi Caps SHIFT + 8 (anche per lo STOP alla riga 90 del programma Autore).
Questa procedura vale anche per le linee 20 dei programmi Autore e Titolo e la 60 di Colloc.
Le parole Coll., Autore, Titolo e Chiavi alla riga 30 di Archivia sono ottenute con Inverse Video all'inizio di ciascuna di esse (e, ovviamente, True Video alla fine di ciascuna).
Infine, ogni volta che compare Erase nel programma Lookfor, anche se

Archivio per Spectrum

Seguito listato Archivio.

```
 120 RESTORE
 130 FOR f=SGN PI TO VAL "7": RE
AD a$: READ d: READ e
 140 PRINT AT NOT PI,SGN PI;"0 P
ER USCIRE";AT VAL "21",VAL "24";
".";AT NOT PI,VAL "29"; BRIGHT S
GN PI;a
 150 IF f=g THEN FLASH SGN PI
 160 PRINT AT f*INT PI-SGN PI,NO
T PI;f;a$;AT f*INT PI,VAL "5"; I
NVERSE SGN PI;b$(a,d TO e): FLAS
H NOT PI: NEXT f: RETURN
```

LOOKFOR

```
E "("
 2 DATA ".Autore",VAL "25",COD
E "$"
 3 DATA ".Titolo",SGN PI,VAL "
24"
 4 DATA "...Anno",CODE ")",COD
E "*"
 5 DATA "...Ch.1",CODE "+",COD
E "7"
 6 DATA "...Ch.2",CODE "8",COD
E "D"
 7 DATA "...Ch.3",CODE "E",COD
E "Q"
 10 INK NOT PI: PRINT " Carica
il blocco"
 20 INK VAL "6": LOAD "arc" DAT
A b$(): INK NOT PI: CLS
 30 FOR a=SGN PI TO CODE "d"
 40 GO SUB VAL "150"
 50 LET g=CODE INKEY$-CODE "0":
IF g<NOT PI OR g>VAL "8" THEN G
O TO CODE "2"
 60 IF g=NOT PI THEN GO TO CODE
 "n"
 70 IF g=VAL "8" THEN GO TO COD
E "d"
 80 RESTORE g: GO SUB CODE " ER
ASE ": GO SUB VAL "160": RESTORE
 g: GO SUB CODE " ERASE ": INPUT
 b$(a,d TO e): GO TO CODE "("
 100 NEXT a
 110 SAVE "arc" DATA b$()
 120 INPUT "Altri blocchi? ";a$
 130 IF CODE a$<>CODE "s" THEN S
TOP
 140 CLS : GO TO VAL "10"
 150 LET g=NOT PI
 160 RESTORE
 170 FOR f=SGN PI TO VAL "7": GO
 SUB CODE " ERASE "
 180 PRINT AT NOT PI,NOT PI;"0 P
ER USC./8 NEXT";AT VAL "21",VAL
"24";".";AT NOT PI,VAL "29"; BRI
GHT SGN PI;a
 190 IF f=g THEN FLASH SGN PI
 200 PRINT AT f*INT PI-SGN PI,NO
T PI;f;a$;AT f*INT PI,VAL "5"; I
NVERSE SGN PI;b$(a,d TO e): FLAS
H NOT PI: NEXT f: RETURN
 210 READ a$: READ d: READ e: RE
TURN
```

tra virgolette, va battuto come comando, non lettera per lettera.
L'insieme dei blocchi è ripetuto diverse volte sul nastro, così da non dover continuamente riavvolgere. Nelle ripetizioni successive l'introduzione (Archivio) è ridotta praticamente al menu per cui l'intervallo che segue può essere ulteriormente ridotto. L'intestazione del blocco introduttivo in questa versione è Archivio1, sempre LINE 130.
Il listato di Archivio1 è identico a quello di Archivio meno le linee 10, 120, 130, 140, 180, e con la seguente modifica:

270 CLS: PRINT INK 0 `` `` `` ``
CORREZIONI " `` `` `` " I LAVORI
IN POSIZIONE NOTA " `` `` 2 LAVORI
DA RICERCARE

I blocchi-dati si registrano su cassette a parte che consigliamo di breve durata; oppure ma la cosa ci sembra meno agevole, è possibile registrare il programma all'inizio della stessa cassetta con i blocchi-dati.
Nulla vieta comunque di utilizzare cassette col nastro lungo - una normale C 60 può contenere agevolmente 4.000 e più titoli.

Si tenga presente che è abbastanza facile e rapido riempire blocchi e blocchi di dati, ma se lo si fa in modo improvvisato è anche facile perdere gran parte dell'utilità di un simile archivio. Innanzitutto è indispensabile assegnare un significato preciso a ciascuna delle posizioni della collocazione. E in fase di immissione si valuti anche in modo appropriato la possibilità che il CUT offre di effettuare la ricerca dei lavori in modo più o meno specifico. Le parole-chiave, che tutte insieme possono essere considerate come un grande indice analitico, sono sicuramente lo strumento più interessante e utile di questo archivio. Tuttavia, mettere delle chiavi a senso, o peggio, a casaccio, magari improvvisando le eventuali abbreviazioni, senza seguire un criterio di massima, riduce sicuramente molto l'efficacia delle successive ricerche per chiave. È senz'altro consigliabile individuare le parole chiave solo sulla base dell'effettivo contenuto (leggere per lo meno l'abstract o il summary), e non solo sulla base del titolo. Molto utile può essere anche un foglietto con le abbreviazioni delle parole-chiave più lunghe e complesse, ripiegato e conservato dentro la stessa scatola della cassetta. Seguendo queste semplici indicazioni, in fase di ricerca il programma consente di individuare in modo rapido e selettivo tutti i lavori che interessano.

Per concludere, volendo puntualizzare i vantaggi offerti dal programma:

- linguaggio BASIC, conosciuto da tutti (anche se il listato si presenta un po' ostico e compresso, in fase operativa la semplicità d'uso è notevole);
- alla portata di chiunque possieda uno Spectrum 16 Kbyte in configurazione semplice (non richiesta stampante o il drive);
- record di adeguata estensione (sulla base della nostra esperienza ci sembra la struttura ottimale: un record più esteso porterebbe spesso a byte in eccesso a quelli necessari, e quindi persi; più limitato porterebbe frequentemente a dover troncare eccessivamente i dati da archiviare, compromettendone la comprensibilità - un discorso simile vale anche per il numero delle parole-chiave/lavoro e per la loro lunghezza);
- infine, riuscire a far maneggiare a uno Spectrum 16 Kbyte simili record in blocchi da 100 non ci sembra poco. ■

SERVIZIO SOFTWARE

PERSONAL SOFTWARE

P.S. propone ai propri lettori i dischi o le cassette dei programmi pubblicati. I programmi, provati e garantiti, sono di immediato utilizzo.

P.S. n°	Programma	Sistema	Prezzo	Codice	Supporto
3	La carta del cielo Collisione	Apple II	30.000	1	Disco
4	Interi in precisione multipla Grafica 3D	Apple II	40.000	4	Disco
5	Pretty printer Shape table	Apple II	30.000	6	Disco
7	Data base modulare	Apple II	25.000	7	Disco
12-13	Wei-ch'i	CBM 3032	20.000	8	Cassetta
14	Tool-Kit	C 64	35.000	9	Cassetta
19	Type Writer	VIC 20	30.000	10	Disco
20	Scopa	C 64 - 3032	25.000	11	Cassetta

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 P.S.

GRUPPO EDITORIALE JACKSON

Cod. ⌊_⌊_⌊_⌊_⌊_⌊_⌋ **a L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

Cod. ⌊_⌊_⌊_⌊_⌊_⌊_⌋ **a L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

Cod. ⌊_⌊_⌊_⌊_⌊_⌊_⌋ **a L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

Cod. ⌊_⌊_⌊_⌊_⌊_⌊_⌋ **a L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

Cod. ⌊_⌊_⌊_⌊_⌊_⌊_⌋ **a L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

Cognome .....

Nome .....

Indirizzo .....

CAP .....

Città .....

SPESE POSTALI (contributo fisso) L. 3.000 **TOTALE L.** ⌊_⌊_⌊_⌊_⌊_⌊_⌋

che pagherò al postino alla consegna del pacco

Firma .....

PERSONAL SOFTWARE

Come aggiungere nuovi comandi BASIC al C 64

Come aggiungere 35 comandi BASIC personalizzati, al nostro Commodore 64

di *Maurizio Paolinelli*

Introduzione

Questo articolo è dedicato in particolare a chi ha già un minimo di cognizione del linguaggio Assembly e sta incominciando ad esplorare quella parte di memoria, troppo spesso ingiustamente considerata tabù, che giace nell'area ROM $A000-$BFFF (Interprete BASIC).
Quando si incomincia a programmare in Assembly nasce, prima o poi, l'esigenza di occuparsi di software di base e di costruire e aggiungere nuovi comandi BASIC personalizzati alla serie dei comandi standard.
Questo problema può essere risolto in modo professionale riprogettando e riscrivendo completamente alcune routine dell'interprete BASIC; ciò richiede però una completa padronanza dell'architettura dell'interprete stesso e un'eccellente confidenza col linguaggio Assembly.

Il problema può anche essere affrontato in modo dilettantesco, ma in compenso alla portata di tutti, seguendo la strada della *sostituzione* di comandi standard poco usati o non indispensabili (come LET, WAIT, VERIFY, CONT, ...) con i nuovi comandi personalizzati; i vari articoli pubblicati sulla creazione di nuovi comandi BASIC (si veda per esempio *Bit* Dicembre '83) impiegano in genere questo procedimento.
Questo è ovviamente un modo di by-passare il problema, ma non di risolverlo, in quanto, essendo estremamente limitato il numero di comandi BASIC non indispensabili o poco utili, il numero dei comandi personalizzati che si possono aggiungere risulta necessariamente altrettanto contenuto.
In questo articolo viene suggerita un'idea che consente di *aggiungere*, e non sostituire, fino a 35 comandi BASIC personalizzati mediante un limitatissimo numero di modifiche all'interprete standard.
Queste modifiche, estremamente semplici, non occupano che 93 byte di memoria; inoltre, le nuove routine relative ai comandi personalizzati non restringono, in genere, l'area della memoria utente, in quanto possono essere memorizzate nell'area fantasma RAM $A000-$BFFF (per l'utilizzo di quest'area si veda, fra l'altro, l'articolo di A. Guida su **Personal Software** n. 18 - Maggio '84, pag. 101).

La realizzazione di questa idea consente dunque di *disporre di 35 comandi BASIC aggiuntivi*, che ciascuno potrà sbizzarrirsi e divertirsi a progettare, *senza restringere*, se si trascurano i 93 byte appena citati, *la memoria a disposizione dell'utente*, che resta definita nelle aree $0800-$9FFF (i classici 38 Kbyte BASIC) e $C05E-$CFFF (3,9 Kbyte invece dei normali 4 Kbyte).

Per poter comprendere e seguire il semplice ragionamento più avanti esposto è necessario conoscere, almeno superficialmente, come le linee BASIC vengono gestite dall'interprete del C 64; il prossimo paragrafo è dedicato a questo argomento e può dunque essere tralasciato dal lettore già esperto.

La gestione delle linee BASIC

Quando si introduce una linea da tastiera, dopo aver premuto il tasto RETURN, viene eseguita la cosiddetta "Warm Start Routine" (puntata dal vettore $0302). Questa routine (WSR nel seguito) per prima cosa carica l'intera linea nell'"input buffer" (si veda anche il diagramma di flusso di figura 1); quindi esegue la cosiddetta "ToKenisation Routine" (TKR).
Quest'ultima confronta successivamente i caratteri nel buffer con i caratteri memorizzati nella "Tabella dei Comandi" (CTB). La CTB, scritta a partire dalla locazione $A09E, contiene in sequenza tutte le parole chiave previste dal BASIC standard (ad esempio END, FOR, - si veda la figura 2).
Se la sequenza dei caratteri nel buffer non corrisponde a nessuno dei nomi contenuti nella tabella, viene segnalato un SYNTAX ERROR: la parola non è prevista dal linguaggio e quindi non interpretabile. Viceversa, se la sequenza corrisponde ad uno dei comandi previsti nella tabella, viene eseguita la tokenizzazione del comando stesso; a seconda della posizione che la parola chiave occupa nella CTB, alla parola stessa viene attribuito un numero a partire da 128 (ad esempio: END = 128 = 1ª posizione in tabella; FOR = 129 = 2ª posizione in tabella; NEW = 162 = 35ª posizione in tabella).

Ad ognuno dei comandi (35 da AND a NEW) corrisponde una routine che contiene la sequenza di istruzioni necessarie per l'esecuzione del comando stesso; ognuna di queste routine sarà caratterizzata dal suo indirizzo, che definisce il punto a partire dal quale, nell'area ROM, è scritta la routine stessa.
A lato della CTB è memorizzata, a

Come aggiungere nuovi comandi BASIC al C 64

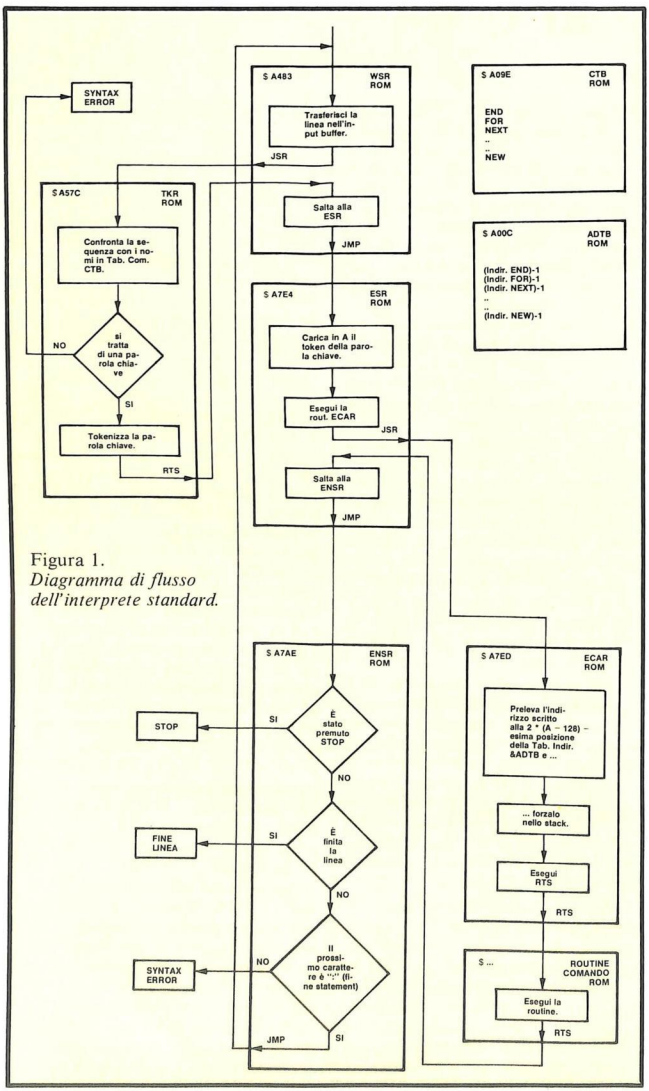


Figura 1.
Diagramma di flusso dell'interprete standard.

partire dalla locazione $A00C, la "Tabella degli Indirizzi" (ADTB; si veda la figura 3) dove, in modo sequenziale e nello stesso ordine con cui i comandi sono memorizzati nella CTB, sono riportati gli indirizzi di partenza delle singole routine relative alle parole chiave.

Dopo aver eseguito la tokenizzazione, la WSR salta alla routine "Esegui uno Statement" (ESR), puntata dal vettore $0308.

Questa routine:

1) carica nell'accumulatore A il codice del comando da eseguire (ad esempio 128 se END, 129 se FOR, ...);

2) esegue la routine "Esegui il Comando in A" (ECAR);

3) salta alla routine "Esegui il Prossimo Statement" (ENSR).

La routine ECAR ha il compito di prelevare dalla ADTB, in funzione del valore memorizzato in A, l'indirizzo della routine interessata (se A = 128, preleva l'indirizzo scritto nelle prime due posizioni della tabella, corrispondente al comando END; se A = 129, preleva l'indirizzo scritto nelle seconde due posizioni della tabella, corrispondente al comando FOR e così via), quindi di saltare ad eseguire la routine scritta all'indirizzo appena prelevato, ovvero di andare a svolgere la sequenza di istruzioni che definisce il comando stesso. Una volta eseguito il comando, il controllo passa alla ENSR che controlla:

1) se è stato premuto il tasto STOP (in tal caso arresta l'esecuzione della linea);

2) se è stata raggiunta la fine della linea (in tal caso è finita l'esecuzione della linea);

3) se è stata raggiunta la fine dello statement (carattere separatore ":").

In quest'ultimo caso si risalta alla WSR per l'esecuzione di un nuovo statement.

Resta soltanto un'ultima precisazione; si sarà notato (figura 3) che la Tabella degli Indirizzi ADTB contiene in realtà l'indirizzo della routi-

Come aggiungere nuovi comandi BASIC al C 64

ne diminuito di uno. Questo è dovuto al modo in cui termina la routine ECAR; essa, infatti, carica nello stack l'indirizzo della tabella e termina con l'istruzione Assembly RTS.

Questa istruzione (RETURN from Subroutine) esegue un salto all'indirizzo contenuto in quel momento in cima allo stack, incrementato di uno. Ovvio dunque che se si è in precedenza forzato nello stack l'indirizzo cui si vuole effettuare il salto meno uno, il salto avverrà all'indirizzo esatto.

L'aggiunta dei nuovi comandi

La soluzione qui proposta per aggiungere comandi personalizzati al BASIC originale si basa sull'idea di impiegare i nomi dei comandi standard, già memorizzati nella Tabella CTB, anche per i nuovi comandi: questi ultimi si distinguono dai primi solo per l'aggiunta, alla fine del nome, del carattere di differenziazione "&".

Unica regola di sintassi da rispettare è che fra il nome e il simbolo "&" non devono essere interposti altri simboli che non siano blank.

Esempio di nomi leciti per un nuovo comando:

END&, FOR&, NEW&,
INPUT#&,...

Questo vincolo nell'attribuzione dei nomi ai nuovi comandi ha l'inconveniente di costringere l'utente a compilare una tabella di equivalenza [vecchio comando + "&"] ≡ nuovo comando, che potrà essere consultata in fase di programmazione (esempio: END& ≡ HPLOT; FOR& ≡ VPLOT; ...).

Si tratta in realtà di un inconveniente molto lieve, se si pensa che lo stesso problema esiste ad esempio per l'abbreviazione dei comandi (? ≡ PRINT, ...), per il codice dei colori (0 ≡ BLACK, ...), ecc.. Esisterà semplicemente una tabella in più da consultare.

A fianco di questo piccolo inconveniente c'è invece il grosso vantaggio di consentire una modifica estremamente semplice dell'interprete standard.

Grazie infatti a questo vincolo, è possibile utilizzare la stessa ToKenisation Routine TKR e la stessa Tabella dei Nomi CTB dell'interprete standard.

È invece necessario sostituire la vecchia ESR (Esegui uno Statement) con la nuova &ESR e scrivere ovviamente una nuova tabella &ADTB

```
                    301  ; table of keyword
                    302  ; same sequence as address table TA00C
                    303  ;
A09E   454EC4       304  TA09E    ● BY 'E, 'N, 'D + $ 80
A0A1   464FD2       305            ● BY 'F, 'O, 'R + $ 80
A0A4   4E4558       306            ● BY 'N, 'E, 'X, 'T + $ 80
A0A8   444154       307            ● BY 'D, 'A, 'T, 'A + $ 80
A0AC   494E50       308            ● BY 'I, 'N, 'P, 'U, 'T, # + $ 80
A0B2   494E50       309            ● BY 'I, 'N, 'P, 'U, 'T + $ 80
A0B7   4449CD       310            ● BY 'D, 'I, 'M + $ 80
A0BA   524541       311            ● BY 'R, 'E, 'A, 'D + $ 80
A0BE   4C45D4       312            ● BY 'L, 'E, 'T + $ 80
A0C1   474F54       313            ● BY 'G, 'O, 'T 'O, + $ 80
A0C5   5255CE       314            ● BY 'R, 'U, 'N + $ 80
A0C8   49C6         315            ● BY 'I, 'F + $ 80
A0CA   524553       316            ● BY 'R, 'E, 'S, 'T, 'O, 'R, 'E + $ 80
A0D1   474F53       317            ● BY 'G, 'O, 'S, 'U, 'B + $ 80
A0D6   524554       318            ● BY 'R, 'E, 'T, 'U, 'R, 'N + $ 80
A0DC   5245CD       319            ● BY 'R, 'E, 'M + $ 80
A0DF   53544F       320            ● BY 'S, 'T, 'O, 'P + $ 80
A0E3   4FCE         321            ● BY 'O, 'N + $ 80
A0E5   574149       322            ● BY 'W, 'A, 'I, 'T + $ 80
A0E9   4C4F41       323            ● BY 'L, 'O, 'A, 'D + $ 80
A0ED   534156       324            ● BY 'S, 'A, 'V, 'E + $ 80
A0F1   564552       325            ● BY 'V, 'E, 'R, 'I, 'F, 'Y + $ 80
A0F7   4445C6       326            ● BY 'D, 'E, 'F + $ 80
A0FA   504F4B       327            ● BY 'P, 'O, 'K, 'E + $ 80
A0FE   505249       328            ● BY 'P, 'R, 'I, 'N, 'T, # + $ 80
A104   505249       329            ● BY 'P, 'R, 'I, 'N, 'T + $ 80
A109   434F4E       330            ● BY 'C, 'O, 'N, 'T + $ 80
A10D   4C4953       331            ● BY 'L, 'I, 'S, 'T + $ 80
A111   434CD2       332            ● BY 'C, 'L, 'R + $ 80
A114   434DC4       333            ● BY 'C, 'M, 'D + $ 80
A117   5359D3       334            ● BY 'S, 'Y, 'S + $ 80
A11A   4F5045       335            ● BY 'O, 'P, 'E, 'N + $ 80
A11E   434C4F       336            ● BY 'C, 'L, 'O, 'S, 'E + $ 80
A123   4745D4       337            ● BY 'G, 'E, 'T + $ 80
A126   4E45D7       338            ● BY 'N, 'E, 'W + $ 80
```

Figura 2. *Tabella dei Comandi CTB (da "Il S.O. del CBM 64" della EVM).*

```
                    203  ; address table for BASIC command
                    204  ; address - 1 used since routine reached via RTS
                    205  ;
A00C   30A8         206  TA00C    ● W  WA831-1    ; END
A00E   41A7         207            ● W  WA742-1    ; FOR
A010   1DAD         208            ● W  WAD1E-1    ; NEXT
A012   F7A8         209            ● W  WA8F8-1    ; DATA
A014   A4AB         210            ● W  WABA5-1    ; INPUT#
A016   BEAB         211            ● W  WABBF-1    ; INPUT
A018   80B0         212            ● W  WB081-1    ; DIM
A01A   05AC         213            ● W  WAC06-1    ; READ
A01C   A4A9         214            ● W  WA9A5-1    ; LET
A01E   9FA8         215            ● W  WA8A0-1    ; GOTO
A020   70A8         216            ● W  WA871-1    ; RUN
A022   27A9         217            ● W  WA928-1    ; IF
A024   1CA8         218            ● W  WA81D-1    ; RESTORE
A026   82A8         219            ● W  WA883-1    ; GOSUB
A028   D1A8         220            ● W  WA8D2-1    ; RETURN
A02A   3AA9         221            ● W  WA93B-1    ; REM
A02C   2EA8         222            ● W  WA82F-1    ; STOP
A02E   4AA9         223            ● W  WA94B-1    ; ON
A030   2CB8         224            ● W  WB82D-1    ; WAIT
A032   67E1         225            ● W  WE168-1    ; LOAD
A034   55E1         226            ● W  WE156-1    ; SAVE
A036   64E1         227            ● W  WE165-1    ; VERIFY
A038   B2B3         228            ● W  WB3B3-1    ; DEF
A03A   23B8         229            ● W  WB824-1    ; POKE
A03C   7FAA         230            ● W  WAA80-1    ; PRINT#
A03E   9FAA         231            ● W  WAAA0-1    ; PRINT
A040   56A8         232            ● W  WA857-1    ; CONT
A042   9BA6         233            ● W  WA69C-1    ; LIST
A044   5DA6         234            ● W  WA65E-1    ; CLR
A046   85AA         235            ● W  WAA86-1    ; CMD
A048   29E1         236            ● W  WE12A-1    ; SYS
A04A   BDE1         237            ● W  WE1BE-1    ; OPEN
A04C   C6E1         238            ● W  WE1C7-1    ; CLOSE
A04E   7AAB         239            ● W  WAB7B-1    ; GET
A050   41A6         240            ● W  WA642-1    ; NEW
```

Figura 3. *Tabella standard degli Indirizzi (da "Il S.O. del CBM 64" della EVM).*

Come aggiungere nuovi comandi BASIC al C 64

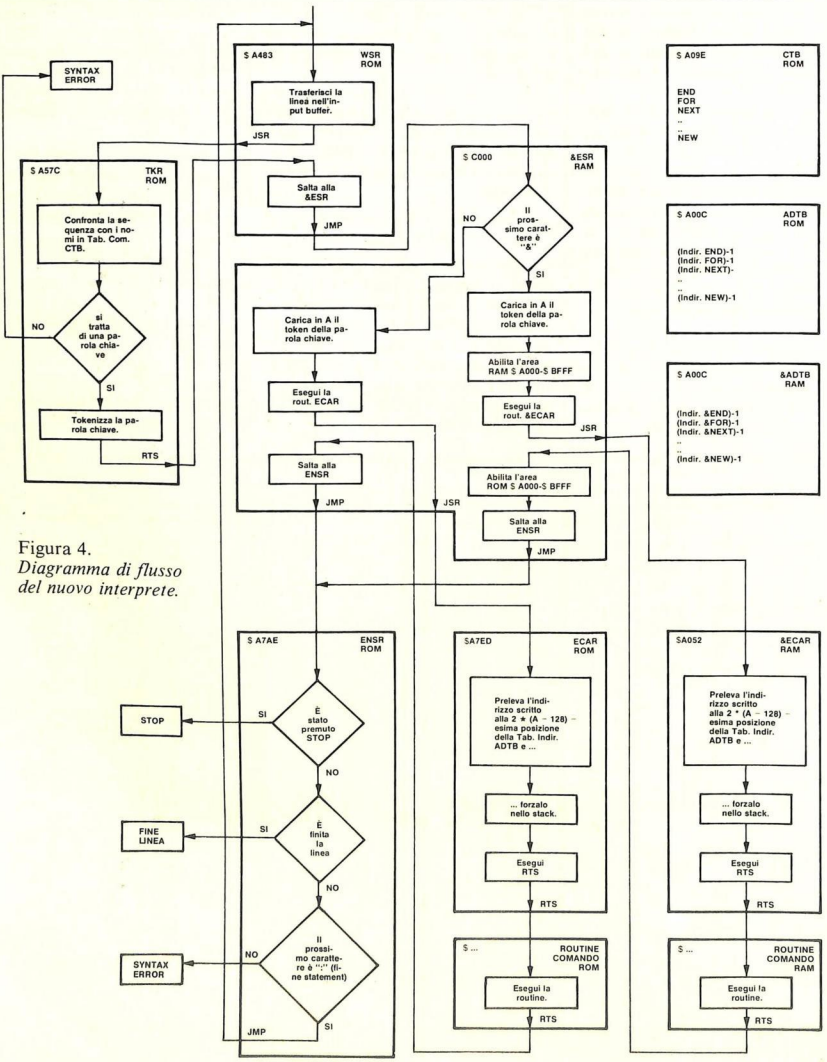


Figura 4.
Diagramma di flusso del nuovo interprete.

Come aggiungere nuovi comandi BASIC al C 64

contenente gli indirizzi delle routine dei nuovi comandi, che si affianchi alla ADTB già esistente in ROM per i comandi originali (si veda il nuovo diagramma di flusso di figura 4).
La nuova &ADTB viene memorizzata nell'area RAM "fantasma" posta in parallelo alla ROM, a partire dallo stesso indirizzo $A00C.

Il compito della nuova &ESR è quello di controllare se al termine dei nomi dei comandi standard sia o no presente il carattere "&"; se la risposta è negativa, si procede come previsto dall'interprete originale; se la risposta è affermativa, si abilita la RAM "fantasma" (e per conseguenza si disabilita la ROM), in modo che la routine ECAR (Esegui il Comando in A) vada a consultare la nuova tabella &ADTB invece della vecchia ADTB e quindi salti ad eseguire il nuovo comando al posto del vecchio.
La ECAR da eseguire in caso di nuovo comando deve risiedere in RAM, essendo disabilitata in tal caso la ROM; per questo motivo la parte fondamentale della ECAR è stata ricopiata in RAM a partire dalla locazione $A052 e compare col nome di &ECAR in figura 4.
Le routine relative ai nuovi comandi possono, come già detto in precedenza, essere scritte nell'area RAM "fantasma", in modo da non limitare in alcun modo la memoria a disposizione dell'utente.

Ultimata l'esecuzione del nuovo comando, la &ESR riabilita l'area ROM (disabilitando per conseguenza la RAM) ed esegue il normale salto alla ENSR (Esegui il Prossimo Statement).
La nuova &ESR è stata memorizzata a partire dalla locazione $C000, in modo che sia indipendente dal banco di memoria selezionato (RAM o ROM). Essendo quest'ultima routine puntata dal vettore $0308, è sufficiente modificarne il contenuto per abilitare il nuovo interprete. La routine scritta a partire dalla locazione $C051 esegue questa abilitazione (si veda il disassemblato in figura 6).

```
                        647; standard warm start routine (BASIC Interpreter)
                        648;
A483    2060A5          649         JSR   SA560     ; get statement into the input buffer
A486    867A            650         STX   Z7A       ; move beginning of line address
A488    847B            651         STY   Z7B       ; to current character address
A48A    207300          652         JSR   X0073     ; get next character of statement
A48D    AA              653         TAX
A48E    F0F0            654         BEQ   BA480     ; skip statement if end of line
A490    A2FF            655         LDX   $FF
A492    863A            656         STX   Z3A       ; set Run/Direct switch to Direct
A494    9006            657         BCC   BA49C     ; if no line number
A496    2079A5          658         JSR   SA579     ; encode keywords in statement
A499    4CE1A7          659         JMP   JA7E1     ; and go execute statement
                        1123; execute next statement
                        1124;
A7AE    202CA8          1125JA7AE   JSR   SA82C     ; check Stop Key
A7B1    A57A            1126        LDA   Z7A
A7B3    A47B            1127        LDY   Z7B       ; get ptr to end of old statement in AY
A7B5    C002            1128        CPY   $02
A7B7    EA              1129        NOP
A7B8    F004            1130        BEQ   BA7BE     ; if not direct mode
A7BA    853D            1131        STA   Z3D       ; store pointer for possible CONT
A7BC    843E            1132        STY   Z3E
A7BE    A000            1133BA7BE   LDY   $00
A7C0    B17A            1134        LDA   (Z7A), Y  ; get last byte
A7C2    D043            1135        BNE   BA807     ; if not 0, check for ":"
A7C4    A002            1136        LDY   $02
A7C6    B17A            1137        LDA   (Z7A), Y  ; get high of pointer to next statement
A7C8    18              1138        CLC
A7C9    D003            1139        BNE   BA7CE
A7CB    4C4BA8          1140        JMP   JA84B     ; 0 = end of program
                        1141;
A7CE    C8              1142BA7CE   INY
A7CF    B17A            1143        LDA   (Z7A), Y  ; move statement # into Z39/Z3A
A7D1    8539            1144        STA   Z39
A7D3    C8              1145        INY
A7D4    B17A            1146        LDA   (Z7A), Y
A7D6    853A            1147        STA   Z3A
A7D8    98              1148        TYA             ; advance pointer over statement header
A7D9    657A            1149        ADC   Z7A
A7DB    857A            1150        STA   Z7A
A7DD    9002            1151        BCC   JA7E1
A7DF    E67B            1152        INC   Z7B
A7E1    6C0803          1153JA7E1   JMP   (X0308)   ; go execute a statement (normally A7E4)
                        1154;
                        1155; execute a statement
                        1156;
A7E4    207300          1157        JSR   X0073     ; get next character
A7E7    20EDA7          1158        JSR   SA7ED     ; execute command in A
A7EA    4CAEA7          1159        JMP   JA7AE     ; go execute next statement
                        1161; execute command in A
                        1162;
A7ED    F03C            1163SA7ED   BEQ   BA82B     ; if Z = 1, return (dummy statement)
A7EF    E980            1164JA7EF   SBC   $80       ; if A < $80
A7F1    9011            1165        BCC   BA804     ; perform LET
A7F3    C923            1166        CMP   $23       ; if A = > A3 (highest command)
A7F5    B017            1167        BCS   BA80E     ; then check for GOTO
A7F7    0A              1168        ASL   A         ; get 2 * (A - 80) as index to table
A7F8    A8              1169        TAY
A7F9    B90DA0          1170        LDA   TA00C +
                                      1,Y           ; move routine address onto stack
A7FC    48              1171        PHA
A7FD    B90CA0          1172        LDA   TA00C, Y
A800    48              1173        PHA
A801    4C7300          1174        JMP   X0073     ; get first char of parameters and go
```

Figura 5. *Disassemblato delle routine standard (da "Il S.O. del CBM 64" della EVM).*

Come aggiungere nuovi comandi BASIC al C 64

Un'ultima precisazione riguarda l'indirizzo da scrivere nella tabella &ADTB in corrispondenza dei nuovi comandi non utilizzati; si tratta dell'indirizzo della routine &SYE, che ha il compito di segnalare un SYNTAX ERROR in caso di chiamata di un nuovo comando inesistente.
La &SYE è stata scritta a partire dalla locazione $C03F.
In definitiva, le operazioni da eseguire per attivare i nuovi comandi possono così riassumersi:
1) scrivere, a partire da $C000, le routine &ESR, &SYE e di abilitazione;
2) riempire la nuova tabella &ADTB con l'indirizzo della routine &SYE;
3) ricopiare, a partire da $A052, la parte essenziale della ECAR per costituire la nuova &ECAR;
4) scrivere le routine relative ai nuovi comandi nell'area RAM $A061-$BFFF;
5) scrivere nella &ADTB l'indirizzo delle nuove routine di cui al punto 4), in corrispondenza dei nuovi comandi;
6) abilitare il nuovo interprete tramite SYS 49233.

Newcom: un esempio di applicazione

Il programma Newcom, scritto in BASIC, è un esempio di come si possano gestire i nuovi comandi.
Questo programma consente di caricare, una dietro l'altra, nell'area RAM $A061-$BFFF le routine relative ai nuovi comandi personalizzati; provvede alle modifiche dell'interprete standard, secondo quanto previsto nel paragrafo precedente e attiva i nuovi comandi stessi.
Ciò che è necessario conoscere sono le routine dei nuovi comandi che devono essere scritte, una alla volta, sotto forma di frasi DATA in coda al programma; ogni volta che si aggiunge una nuova routine, il pro-

```
                                Routine &ESR - Esegui uno statement (vector addr. $0308)
C000  JSR   $0073               Prendi il prossimo carattere
C003  PHP                       e salvalo, insieme ai "flags",
C004  PHA                       nello stack (codice comando).
C005  LDX   $7A                 Indirizzo carattere corrente
C007  LDY   $7B                 in X, Y.
C009  JSR   $0073               Prendi il prossimo carattere
C00C  CMP   # $26               e confrontalo con "&".
C00E  BEQ   $C01C               Se = "&" (nuovo comando) vai a ... $C01C
C010  STX   $7A                 altrimenti ripristina in $7A, $7B
C012  STY   $7B                 l'indirizzo del carattere precedente;
C014  PLA                       riprendi il codice comando e
C015  PLP                       i relativi "flags";
C016  JSR   $A7ED               esegui il comando in A;
C019  JMP   $A7AE               esegui il prossimo statement.

C01C  PLA                       Riprendi il codice comando e
C01D  PLP                       i relativi "flags".
C01E  JSR   $C02A               Abilita RAM da $A000 a $BFFF.
C021  JSR   $A052               Esegui il comando in A.
C024  JSR   $C038               Abilita ROM da $A000 a $BFFF.
C027  JMP   $A7AE               Esegui il prossimo statement.

                                Subroutine abilitazione RAM
C02A  PHA                       Salva l'accumulatore nello stack.
C02B  SEI                       Disabilita Interrupt.
C02C  LDA   $01                 Salva il contenuto della porta I/O
C02E  STA   $C045               in $C045.
C031  AND   # $FE               Azzera il bit "0" dell'accumulatore.
C033  STA   $01                 Memorizza A nella porta I/O.
C035  CLI                       Riabilita Interrupt.
C036  PLA                       Riprendi il contenuto dell'accumulatore.
C037  RTS                       Ritorna.

                                Subroutine abilitazione ROM
C038  PHA                       Salva l'accumulatore nello stack.
C039  SEI                       Disabilita Interrupt.
C03A  LDA   $C045               Recupera il valore originale della porta I/O
C03D  BNE   $C033               e salta a ... $C033.

                                Routine &SYE (SYNTAX ERROR)
C03F  JSR   $C038               Abilita ROM da $A000 a $BFFF
C042  JMP   $AF08               e ... SYNTAX ERROR.

C045  00                        Registro memor. contenuto porta I/O.

                                Routine abilitazione nuovo interprete
C051  SEI                       Disabilita Interrupt.
C052  LDA   # $00
C054  STA   $0308
C057  LDA   # $C0               $C000 in $0308, $0309.
C059  STA   $0309
C05C  CLI                       Riabilita Interrupt.
C05D  RTS                       Ritorna.

                                Routine &ECAR - Esegui comando in A (RAM)
A052  SBC   # $80               Sottrai 128 al codice in A;
A054  ASL                       moltiplica per 2 (A = 2 * (A - 128)).
A055  TAY
A056  LDA   $A00D, Y            Carica nello stack
A059  PHA                       l'indirizzo del comando
A05A  LDA   $A00C, Y            da eseguire.
A05D  PHA
A05E  JMP   $0073               Prendi il prossimo carattere ed esegui.
```

Figura 6. *Disassemblato delle nuove routine.*

gramma va ovviamente salvato su supporto esterno, affinché il nuovo comando resti memorizzato nel programma.
Le frasi DATA relative alle routine dei nuovi comandi devono rispettare alcune regole:
1) la prima informazione DATA *deve* essere il nome mnemonico del nuovo comando (massimo 7 caratteri, primo carattere letterale);
2) l'ultima informazione DATA *deve* essere il carattere "★".
Anche il modo di scrivere le routine deve soggiacere ad alcuni vincoli, affinché i nuovi comandi possano essere gestiti dal Newcom.

Dato che non si conosce a priori dove le varie routine saranno allocate, queste dovranno essere scritte in modo da poter essere rilocate. Inoltre tutte le routine devono tassativamente terminare con un RTS.
Alcuni spunti per la creazione di comandi relativi alla grafica in alta risoluzione si possono trovare in *Bit* Dicembre '83.
Un esempio di nuovo comando, che consente l'emissione di un impulso sonoro, è inoltre riportato in fondo al Newcom sotto il nome di BEEP. Ciascuno potrà inventare e progettare i comandi che riterrà più idonei per soddisfare le proprie esigenze.
A questo punto non resta che digitare il programma e lanciarlo; la sua esecuzione e i pochi commenti che seguono dovrebbero essere più che sufficienti per la comprensione del suo funzionamento.

Inizialmente vengono eseguite le modifiche all'interprete standard descritte ai punti 1), 2), 3) del paragrafo precedente.
Viene poi visualizzata la tabella del-

Listato 1. *Il programma principale.*

```
0 REM ********************************
1 REM *     ----------------------     *
2 REM **** NUOVI COMANDI 'BASIC' ****
3 REM ****         PER C.64        ****
4 REM *     ----------------------     *
5 REM *                DI              *
6 REM * MAURIZIO PAOLINELLI            *
7 REM * VIA MAGELLANO,15               *
8 REM * 20094 CORSICO  MI              *
9 REM * TEL. (02)4407707               *
10 REM*                                *
11 REM********************************
18 POKE53281,14:PRINT"[<1BLK>]":PRINT"[<1
CLR>]"SPC(17)"NEWCOM[<6CRSR D>]"
19 PRINTTAB(12)"ATTENDI, PREGO!"
20 DIMCN$(35),CS$(35),A(35),AD(35)
25 FORI=1TO35:READCS$(I):NEXT:REM***
                  LEGGI NOMI COMANDI STD.
51 REM***SPAZIO DISPONIBILE PER LA DEF.
85 REM***DEL CODICE NUOVI COM.[A(I)=1,...
35]
100 FORI=1TO35:IFA(I)=0THENN=I-1:I=35
105 NEXT:REM***N=N.RO COMANDI IN OPERA
110 IFN=0THENPOKE836,97:POKE837,160:GOTO1
70
114 REM***SCRIVI NUOVA ROUTINE 'ESEGUI
        UNO STATEMENT'.
115 FORI=49152TO49245:READM:POKEI,M:NEXT
119 REM***SCRIVI NUOVA ROUTINE 'ESEGUI
        COMANDO IN A'
120 POKE41042,233:POKE41043,128
125 FORI=42999TO43011:POKEI-1955,PEEK(I):
NEXT
129 REM***SCRIVI ROUTINES NUOVI COMANDI
        DALLA LOCAZ. 41057
130 I=41056:AD(1)=I+1:FORH=1TON
134 REM***LEGGI NOME NUOVO COMANDO
135 READCN$(A(H)):IFASC(LEFT$(CN$(A(H)),1
))<65THEN410
140 I=I+1:READA$:IFA$<>"*"THENPOKEI,VAL(A
$):GOTO140
145 AD(H+1)=I:I=I-1:NEXT
149 REM***RIEMPI NUOVA TAB. INDIRIZZI
        CON INDIRIZZO SYNTAX ERROR
150 FORI=40972TO41040STEP2:POKEI,62:POKEI
+1,192:NEXT
154 REM***DEF.INDIRIZZI NUOVE ROUTINES
        NELLA NUOVA TAB. INDIRIZZI
155 FORI=1TON:HB=INT((AD(I)-1)/256):LB=AD
(I)-1-256*HB
160 AC=40970+2*A(I):POKEAC,LB:POKEAC+1,HB
:NEXT
162 HB=INT((AD(N+1))/256):LB=AD(N+1)-256*
HB:POKE836,LB:POKE837,HB
164 REM***ABILITA NUOVA ROUT. 'ESEGUI
        UNO STATEMENT'
165 SYS49233:M=0
169 REM***STAMPA ELENCO NUOVI COMANDI
170 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<2CRSR
D>]"
175 FORI=1TO17:IFCN$(I)=""THENCN$(I)="...
"
180 IFCN$(I+18)=""THENCN$(I+18)="..."
185 PRINTMID$(STR$(I),2,2);TAB(2)"."CS$(I
)"&";
190 PRINTTAB(11)"_"LEFT$(CN$(I),7);TAB(21
);
195 PRINTRIGHT$(STR$(I+18),2)"."CS$(I+18)
"&";
200 PRINTTAB(31)"_"LEFT$(CN$(I+18),7):NEX
T:IFCN$(18)=""THENCN$(18)="..."
205 PRINT"18.ON&      "LEFT$(CN$(18),7)
210 PRINT"[<1CRSR D>]NUMERO COMANDO DA AG
GIUNGERE"
215 INPUT"(0 PER NON AGGIUNGERE COM.)";C:
REM***C=COD. NUOVO COMANDO
220 IFC=0ANDN=0THEN360:REM***NESSUN NUO-
                              VO COMANDO
225 IFC=0THEN375:REM***NUOVI COMANDI
                        ATTIVATI
230 IFC<0ORC>35THENPRINT"[<12CRSR L>]
     [<4CRSR U>]":GOTO210
235 FORI=1TON:IFA(I)=CTHENM=I
240 NEXT:REM***M=COD. COM. DA SOST.
245 IFM=0THEN385:REM***CODICE LIBERO
250 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
255 PRINT"STAI PER SOSTITUIRE IL COMANDO:
[<1CRSR D>]"
```

Seguito listato 1.

```
260 PRINTMID$(STR$(M),2,2);TAB(2)"."CS$(A
(M))"&  _  "LEFT$(CN$(A(M)),7)
265 INPUT"[<2CRSR D>]SEI D'ACCORDO (S/N)"
;A$
270 IFA$<>"S"ANDA$<>"N"THEN265
275 IFA$="N"THEN170
276 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
277 PRINTTAB(12)"ATTENDI, PREGO!"
279 REM***CANCELLA I 'DATA' DEL COMANDO
        DA SOSTITUIRE
280 POKE834,M:POKE835,C:RESTORE
281 HB=INT((AD(M))/256):LB=AD(M)-256*HB:P
OKE836,LB:POKE837,HB
285 READA$:IFA$<>CN$(A(M))THEN285
290 POKE832,PEEK(63):POKE833,PEEK(64)
293 REM***L1=PRIMA LINEA DA CANCELL.
        N1=ADDRESS NEXT LINE
294 REM***L2=PRIMA LINEA DA SALVARE
295 L1=PEEK(63)+256*PEEK(64):N1=PEEK(65)+
256*PEEK(66)
296 I=LEN(CN$(A(M)))
297 IFPEEK(N1-I)<>131THENI=I+1:GOTO297
298 N1=N1-I-5
300 HB=INT(N1/256):LB=N1-256*HB
305 POKE828,LB:POKE829,HB:IFM=NTHENPOKE83
0,0:POKE831,0:GOTO320
310 READA$:IFA$<>CN$(A(M+1))THEN310
315 POKE830,PEEK(63):POKE831,PEEK(64)
320 PRINT"[<1CLR>][<2CRSR D>]"L1:PRINT"GO
TO 330[<1HOME>]";
325 POKE631,13:POKE632,13:POKE198,2:END:R
EM***CANCELLA
330 N1=PEEK(828)+256*PEEK(829):L1=PEEK(N1
+3)+256*PEEK(N1+4)
335 L2=PEEK(830)+256*PEEK(831):IFL1=L2THE
N342
340 GOTO320
342 M=PEEK(834):C=PEEK(835):GOTO500
345 L1=PEEK(832)+256*PEEK(833):L2=PEEK(83
0)+256*PEEK(831)
350 GOSUB450:PRINT"[<1RVS>]3.[<1RVS OFF
>] I NUMERI DI LINEA UTILI SONO"
352 L2$=STR$(L2-1):IFL2=0THENL2$=" - "
355 PRINT"    DA"L1"A"L2$".":GOSUB485:END
359 REM***NESSUN NUOVO COMANDO
360 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
365 PRINT"NESSUN NUOVO COMANDO ATTIVATO!"
370 PRINT"RILANCIA IL PROGRAMMA CON UN 'R
UN'.":END
374 REM***NUOVI COMANDI ATTIVATI
375 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
380 PRINT"NUOVI COMANDI ATTIVATI!":NEW:EN
D
384 REM***AGGIUNTA NUOVO COMANDO
385 GOTO510
387 GOSUB450:GOSUB485
390 PRINT"[<3CRSR D>]PREMI 'RETURN' PER C
ONTINUARE":POKE198,0
395 GETA$:IFA$<>CHR$(13)THEN395
```

COMPUSCUOLA

La rivista di informatica
nella didattica per la scuola

Con tutta la competenza del
GRUPPO EDITORIALE
JACKSON

Come aggiungere nuovi comandi BASIC al C 64

Seguito listato 1.

```
400 PRINT"[<1CLR>]":LIST650-:END
410 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
415 PRINT"NOME NON AMMESSO PER LA ROUTINE
 "CN$(A(H))
420 PRINT"[<1CRSR D>]IL PRIMO CAR. DEVE E
SSERE LETTERALE.":END
449 REM***ISTRUZIONI
450 PRINT"[<1CLR>]"SPC(17)"NEWCOM[<6CRSR
D>]"
455 PRINT"AGGIUNGI I 'DATA' DELLA SUBROUT
INE DEL"
460 PRINT"NUOVO COMANDO."
465 PRINT"[<1RVS>]1.[<1RVS OFF>] IL PRI
MO 'DATA' DEVE ESSERE"
470 PRINT"    IL NOME DEL NUOVO COMANDO"
475 PRINT"    (7 CAR.MAX.- 1[<1CHR$(190)>]
CAR.LETTERA)."
480 PRINT"[<1RVS>]2.[<1RVS OFF>] L'ULTI
MO DATA DEVE ESSERE UN '*'.":RETURN
485 PRINT"[<1CRSR D>][<1RVS>]RICORDATI D
I SALVARE IL PROGRAMMA[<1RVS OFF>]"
490 PRINT"[<1RVS>]PRIMA DI FARLO GIRARE.
[<1RVS OFF>]"
492 R=PEEK(836)+256*PEEK(837)
495 PRINT"[<1CRSR D>]LA NUOVA ROUTINE SAR
A' MEMOR.":PRINT"DALL'INDIRIZZO"R".":RETU
RN
500 PRINT"[<1CLR>][<2CRSR D>]"M+50"A("MID
$(STR$(M),2,2)")="MID$(STR$(C),2,2)
505 PRINT"GOTO 345[<1HOME>]";:POKE631,13:
POKE632,13:POKE198,2:END
510 PRINT"[<1CLR>][<2CRSR D>]"N+51"A("MID
$(STR$(N+1),2,2)")="MID$(STR$(C),2,2)
515 PRINT"GOTO 387[<1HOME>]";:POKE631,13:
POKE632,13:POKE198,2:END
549 REM***ELENCO COMANDI STD
550 DATAEND,FOR,NEXT,DATA,INPUT#,INPUT,DI
M,READ,LET,GOTO,RUN,IF,RESTORE,GOSUB
560 DATARETURN,REM,STOP,ON,WAIT,LOAD,SAVE
,VERIFY,DEF,POKE,PRINT#,PRINT,CONT
570 DATALIST,CLR,CMD,SYS,OPEN,CLOSE,GET,N
EW
599 REM***ROUTINE NEWCOM
600 DATA32,115,0,8,72,166,122,164,123,32,
115,0,201,38,240,12,134
610 DATA122,132,123,104,40,32,237,167,76,
174,167,104,40,32,42,192,32
620 DATA82,160,32,56,192,76,174,167,72,12
0,165,1,141,69,192,41,254
630 DATA133,1,88,104,96,72,120,173,69,192
,208,244,32,56,192,76,8
640 DATA175,55,234,32,56,192,32,210,255,3
2,42,192,96,120,169,0,141
650 DATA8,3,169,192,141,9,3,88,96
1000 DATA        BEEP
1001 DATA169,0,141,4,212,141,5,212,141,6,
212,141,2,212,169,7,141
1002 DATA3,212,169,34,141,5,212,169,82,14
1,1,212,169,65,141,4,212
1003 DATA169,15,141,24,212,96,*
READY.
```

le equivalenze [vecchio comando + "&"] ≡ nuovo comando, di cui si è parlato in precedenza, con la situazione aggiornata dei nuovi comandi già contenuti nel Newcom (la prima volta che il programma verrà caricato e lanciato la tabella non mostrerà alcun nuovo comando).

L'utente viene poi invitato a scegliere il codice numerico da 1 a 35, e per conseguenza il nome, da attribuire al nuovo comando che si vuole immettere (esempio: 1 = END&; 2 = FOR&; ...).

Se il numero è compreso fra 1 e 35, il programma si automodifica, memorizzando, nelle linee da 51 a 85, la presenza del nuovo comando; quindi si arresta, visualizzando la serie di istruzioni da eseguire per la memorizzazione completa del nuovo comando.

Si possono verificare due casi.

Caso a) - Il comando prescelto non era ancora utilizzato in precedenza (si *aggiunge* un nuovo comando).

Le istruzioni avvertono che è necessario:

- aggiungere dopo l'ultima linea di programma la sequenza dei DATA corrispondente alla routine del nuovo comando;
- salvare il programma su supporto esterno prima di lanciarlo.

Caso b) - Il comando prescelto era già utilizzato in precedenza (si *sostituisce* un nuovo comando ad un altro).

Le istruzioni sono analoghe a quelle del caso a), ma c'è un vincolo da rispettare:

- i numeri di linea delle frasi DATA relative al nuovo comando devono essere compresi fra un minimo e un massimo, indicati dalle istruzioni stesse.

N.B. - Con le istruzioni viene comunicato anche l'indirizzo a partire dal quale verrà memorizzata la routine del nuovo comando.

Se l'utente non vuole aggiungere nuovi comandi, ma semplicemente attivare quelli già contenuti nel Newcom, deve rispondere col numero 0 alla domanda di scelta del codice.

In tal caso Newcom memorizza in sequenza le routine già registrate, a partire dall'indirizzo $A061, aggiorna la nuova tavola degli indirizzi &ADTB e quindi attiva i nuovi comandi mediante l'abilitazione del nuovo interprete.

Infine, dopo aver annunciato l'attivazione dei nuovi comandi, il programma si autocancella mediante l'istruzione NEW, liberando completamente l'area BASIC a disposizione dell'utente.

Il Newcom vuole solo essere un esempio abbastanza semplice di come si possano gestire i nuovi comandi: ognuno potrà naturalmente scriversi un programma che possa fare di più e di meglio. ■

PERSONAL SOFTWARE

Impariamo il linguaggio macchina con il VIC 20 e il C 64

— Parte ottava —

Le ultime istruzioni e un programma esemplificativo

di *Alessandro Guida*

Abbiamo già visto in precedenza come si effettua l'addizione in linguaggio macchina e quale è l'istruzione necessaria (ADC).
Inoltre, si è visto che è possibile sommare numeri a 16 bit grazie al riporto (C-carry) che viene automaticamente aggiunto ad ogni addizione. È perciò necessario azzerarlo alla prima somma. Ad esempio, per sommare due numeri a 16 bit conservati nelle locazioni $FB, FC e $FD,FE avremo il seguente programmino:

```
CLC
LDA $FB : somma byte basso
ADC $FD
STA $FD
LDA $FC : somma byte alto
ADC $FE
STA $FE
```

Nella seconda addizione un eventuale riporto viene aggiunto nel byte alto, mentre nella prima ciò non avviene a causa del CLC (azzera carry).
Come per l'addizione esiste un comando per la sottrazione. Il simbolo mnemonico è SBC e opera una sottrazione con carry. Il suo modo di operare è molto simile a quello della addizione: infatti, il 6502 non fa altro che aggiungere al primo operando il complemento a due del secondo più la negazione del carry.
Quindi, se il carry è uno vi aggiunge zero, e viceversa.

È importante perciò ricordarsi che il carry va sempre posto a uno (SEC) prima di una sottrazione. L'esempio precedente diventa:

```
SEC
LDA $FB : sottrae byte basso
SBC $FD
STA $FD
LDA $FC : sottrae byte alto
SBC $FE
STA $FE
```

Abbiamo quindi visto due nuove istruzioni, SBC e SEC, che non presentano particolari difficoltà. Purtroppo, invece, non ci sono istruzioni dirette all'esecuzione delle moltiplicazioni.
Queste vanno calcolate per mezzo di algoritmi più o meno complessi, secondo il tipo di moltiplicazione richiesto. Il caso più semplice si ha moltiplicando due numeri interi ad

```
Tabella Codici Istruzioni

Istruzione: SEC - Pone il Carry a uno
            SBC - Sottrae all'accumulatore l'operando e il C
            JSR - Esegue una subroutine
            ASL - Sposta i bit a sinistra
            LSR - Sposta i bit a destra
            ROL - Ruota i bit a sinistra attraverso il Carry
            ROR - Ruota i bit a destra attraverso il Carry


FLAG alterati:      N V B D I Z C
                SEC               X
                SBC   X X         X X
                JSR
                ASL   X           X X
                LSR   0           X X
                ROL   X           X X
                ROR   X           X X


Modo indirizzamento       Codici
                          SEC SBC JSR ASL LSR ROL ROR

Implicato                 38
Accumulatore                              0A  4A  2A  6A
Assoluto                      ED  20      0E  4E  2E  6E
Pagina-0                      E5          06  46  26  66
Immediato                     E9
Assoluto,X                    FD          1E  5E  3E  7E
Assoluto,Y                    F9
(IND,X)                       E1
(IND),Y                       F1          16  56  36  76
Pagina-0,X                    F5
Pagina-0,Y
Relativo
Indiretto
```

Tabella 1. *Codici e modi di indirizzamento delle istruzioni SEC, SBC, SSR, ASL, LSR, ROL e ROR.*

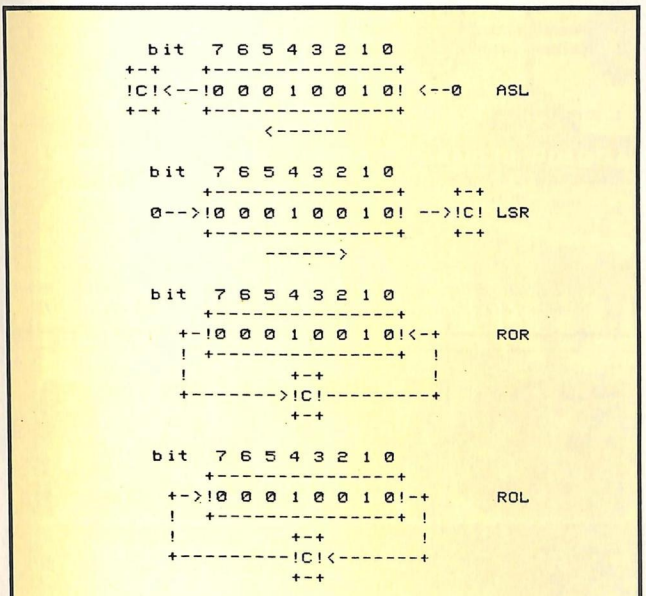


Figura 1. *Funzionamento dei comandi ASL, LSR, ROL e ROR.*

otto bit. In questo caso, si può ridurre l'addizione a un ciclo di somme. Se, ad esempio, si vuole moltiplicare il numero contenuto nella locazione $FB per quello esistente nella locazione $FD avremo:

```
C000 LDA #$00 :   azzera byte alto risultato
C002 STA $FC
C004 LDX $FB  :   valore moltiplicando
C006 LDY $FD  :   contatore ciclo=moltiplicatore
C008 BEQ $C01C:   moltiplicatore = 0
C00A DEY      :   decrementa contatore
C00B BEQ $C022:   termine moltiplicazione
C00D CLC
C00E TXA
C00F ADC $FB  :   somma al risultato il moltiplicando
C011 STA $FB
C013 LDA #$00
C015 ADC $FC  :   somma l'eventuale riporto
C017 STA $FC
C019 JMP $C00A:   ripete il ciclo
C01C LDA #$00 :   se moltiplicatore=0
C01E STA $FB  :   risultato=0
C020 STA $FC
C022 RTS      :   fine operazione torna al BASIC.
```

Viene utilizzata anche la locazione $7C, poiché il prodotto di due numeri ad otto bit può richiederne due per il risultato.
Per dividere due numeri interi si opererebbe in modo analogo, sottraendo invece di aggiungere. Vi è, tuttavia, un caso di moltiplicazione o divisione ancora più semplice. Moltiplicare e dividere per potenze di due.
Basta osservare il seguente esempio: prendiamo il numero 5 in binario

0000 0101 (5 decimale)

e facciamo scorrere tutti i bit a sinistra di un posto, spostando l'ultimo nel carry ed aggiungendo uno zero a sinistra, avremo:

0 0000 1010 (10 decimale)

abbiamo ottenuto una moltiplicazione per 2. Ripetiamo l'esperimento per due volte consecutive:

0 0001 0100 (20 decimale)
0 0010 1000 (40 decimale)

abbiamo moltiplicato altre due volte per due. Nel complesso spostando per tre volte a sinistra il numero iniziale, abbiamo avuto una moltiplicazione per 8, ossia 2↑3 (due alla terza).

In generale si ha che per moltiplicare un numero per una potenza di due basta fare scorrere i bit di tale numero a sinistra di tanti posti quanto il valore dell'esponente del due.
Per dividere basta far scorrere i singoli bit a destra.
Esempio:

0010 0000 (48 decimale)
0001 1000 (24 decimale)

Queste due operazioni di spostamento dei bit a sinistra e a destra sono svolte dalle istruzioni ASL (Spostamento a sinistra) e LSR (Spostamento a destra).
Tutto procede perfettamente finché non si supera il valore 255. Se si moltiplicasse il numero 136 per 2 si avrebbe:

1000 1000 (136 decimale)
1 0001 0000 (16 decimale)
Il risultato è chiaramente sbagliato, poiché il bit più significativo è stato spinto nel carry. Il risultato richiede ora due byte. Infatti, se considerassimo il carry come il primo bit di un secondo byte avremmo 256+16=272, cioé il giusto risultato.
Fortunatamente vi sono le istruzioni ROL (rotazione a sinistra) e ROR (rotazione a destra) che operano come l'ASL e LSR ma introducendo nel primo o ultimo bit invece di uno zero il contenuto del carry.
Per esempio l'operazione 240 x 8 diventerebbe in Assembly:

```
C000 LDA #$F0 :  memorizza #240
C002 STA $FB  :  nelle locazioni
                 $FB,FC
C004 LDA #$00 :  per contenere il risul-
                 tato
```

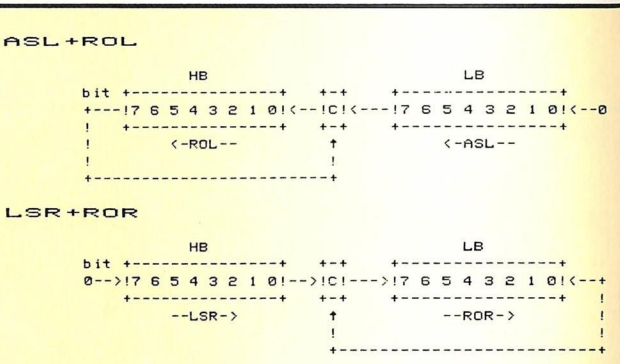


Figura 2. *Effetto delle operazioni ASL+ROL e LSR+ROR su numeri a sedici bit. Se dopo il ROR il carry è uguale a uno, vuol dire che si è avuto un overflow.*

E' IN EDICOLA

BIT

Bit,
la prima rivista europea
di personal computer,
software, accessori,
la più prestigiosa
e più diffusa in Italia

con tutta la competenza del

GRUPPO EDITORIALE JACKSON

È IN EDICOLA

VIDEO GIOCHI 22

GENNAIO
1985
L. 8.000

LA PRIMA E[illegible]VISTA ITALIANA DI VIDEOGAMES E GIOCOMPUTER

Spedizione in Abb. Postale Gruppo III/70

CARATTERISTICHE
E PREZZI DI OLTRE
500 GIOCHI · ACCESSORI ·
COMPUTER

TUTTI
I VIDEOGAMES

```
C006 STA $FC   :  a 16 bit
C008 LDY #$03  :  contatore spostamenti
C00A ASL $FB   :  moltiplica per due byte basso
C00C ROL $FC   :  moltiplica per due il byte alto
C00E DEY       :  ripete tre volte
C00F BNE $C00A
C011 RTS
```

Le subroutine

Per evitare di ripetere molte volte le stesse sequenze di comandi il BASIC dispone del GOSUB-RETURN. Il linguaggio macchina presenta lo stesso comando con il nome JSR (Jump SubRoutine). Il JSR deve essere sempre seguito dall'indirizzo di inizio della subroutine. Al termine della routine deve esservi il RTS (ReTurn from Subroutine) che si era già utilizzato per il ritorno al BASIC.

Quando il microprocessore incontra l'istruzione JSR conserva il contenuto del contatore di programma nello stack e vi forza l'indirizzo della subroutine. Al termine viene ripreso il vecchio indirizzo dallo stack, rimesso nel contatore di programma e incrementato per passare all'istruzione successiva.

Quasi tutte le routine che compongono il sistema operativo del C 64 e VIC 20, essendo chiamate in più punti, sono organizzate come subroutine con un RTS finale. Ad esempio, la subroutine $FFD2 provvede a inviare il carattere contenuto nell'accumulatore alla periferica di output (di solito lo schermo). Così il seguente programmino stampa tutti i caratteri ASCII.

```
C000 LDY #$20  :  primo carattere
C002 TYA       :  carattere da trasmettere
C003 JSR $FFD2:   esegue la subroutine
                  di stampa
C006 INY       :  carattere successivo
C007 BPL $C002 :  se <128 continua
C009 RTS       :  torna al BASIC
```

Indirizzamenti indiretti

Come abbiamo più volte sottolineato, il pregio del linguaggio macchina è senz'altro la sua velocità. È, quindi, giusto che venga applicato

Listato 1. *Programma per lo scroll dello schermo al contrario.*

```
c000 a9 07      lda #$07     :Memorizza in $FB,FC e $FD,FE
c002 85 fc      sta $fc      :gli indirizzi di partenza
c004 85 fe      sta $fe      :della penultima ed ultima
c006 a9 98      lda #$98     :linea della mappa dello schermo
c008 85 fb      sta $fb      :
c00a a9 c0      lda #$c0     :
c00c 85 fd      sta $fd      :
c00e a2 19      ldx #$19     :Numero di linee dello schermo
c010 20 40 c0   jsr $c040    :Routine di scroll
c013 a9 db      lda #$db     :Memorizza gli indirizzi per la
c015 85 fc      sta $fc      :mappa dei colori
c017 85 fe      sta $fe      :
c019 a9 98      lda #$98     :
c01b 85 fb      sta $fb      :
c01d a9 c0      lda #$c0     :
c01f 85 fd      sta $fd      :
c021 a2 19      ldx #$19     :Numero linee
c023 20 40 c0   jsr $c040    :Routine di scroll
c026 a0 27      ldy #$27     :Riempie la prima linea dello
c028 a9 20      lda #$20     :dello schermo di spazi
c02a 99 00 04   sta $0400,y  :
c02d 88         dey          :
c02e 10 fa      bpl $c02a    :
c030 ad 21 d0   lda $d021    :Legge colore sfondo
c033 a0 27      ldy #$27     :Riempie prima linea mappa colori
c035 99 00 d8   sta $d800,y  :col colore dello sfondo
c038 88         dey          :
c039 10 fa      bpl $c035    :
c03b 60         rts          :Torna al Basic
c03c ea         nop          :
c03d ea         nop          :
c03e ea         nop          :
c03f ea         nop          :
c040 a0 27      ldy #$27     :Numero colonne
c042 b1 fb      lda ($fb),y  :Legge carattere linea superiore
c044 91 fd      sta ($fd),y  :Lo copia nella linea inferiore
c046 88         dey          :Carattere successivo
c047 10 f9      bpl $c042    :
c049 38         sec          :Decrementa i puntatori delle due
c04a a5 fb      lda $fb      :linee
c04c e9 28      sbc #$28     :
c04e 85 fb      sta $fb      :
c050 a5 fc      lda $fc      :
c052 e9 00      sbc #$00     :
c054 85 fc      sta $fc      :
c056 38         sec          :
c057 a5 fd      lda $fd      :
c059 e9 28      sbc #$28     :
c05b 85 fd      sta $fd      :
c05d a5 fe      lda $fe      :
c05f e9 00      sbc #$00     :
c061 85 fe      sta $fe      :
c063 ca         dex          :Decrementa numero linea
c064 d0 da      bne $c040    :Se non ha finito copia un'altra
c066 60         rts          :linea
```

Libri firmati JACKSON

Gabriele Ugolini
PERSONAL GRAPHICS

Vengono presentati e descritti un buon numero di programmi di grafica e animazione su APPLE.
La trattazione inizia con i "mattoni" del disegno, punti, linee e superfici, prosegue con l'osservazione di figure nel piano e oggetti nello spazio e la raffigurazione realistica di complesse figure geometriche, sfiora la grafica del video gioco e si conclude con l'animazione stile Walt Disney.

158 pagine

Codice **555D L. 22.000**

Michel Benelfoul
METODI DI REALIZZAZIONE DEI PROGRAMMI

Destinato agli utenti di personal con una certa pratica di programmazione BASIC, il libro propone un metodo per la realizzazione dei programmi.
Per smitizzare il "dialetto" dell'informatica comincia col fornire un glossario completo e rigoroso.
È poi affrontato il problema dell'analisi di un sistema, con particolare attenzione al metodo di scelta dell'hardware, alle possibili riorganizzazioni del sistema ed alle esigenze di elaborazione dei dati.
Col supporto di modelli basati sulla matematica moderna e sulla teoria degli insiemi si affronta infine il problema dell'organizzazione logica dei dati.

96 pagine
Codice **401H L. 10.000**

Reinhold Thurner
PROGRAMMAZIONE STRUTTURATA
Corso di autoistruzione

Questo corso di autoistruzione insegna i principi fondamentali della programmazione strutturata, principi che sono comuni a tutte le effettive applicazioni di questa importante metodologia. Esso è concepito per aiutare il lettore a capire i costrutti ed a applicare correttamente le tecniche della programmazione strutturata. Spiega anche e insegna l'uso corretto delle principali tecniche di rappresentazione strutturata usata in analisi e programmazione (diagrammi di flusso, struttogrammi, pseudocodice e diagrammi ad albero).

136 pagine

Codice **503A L. 13.500**

La Biblioteca che fa testo

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca _____ n° _____

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ 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

GRUPPO EDITORIALE JACKSON

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

nuovidea

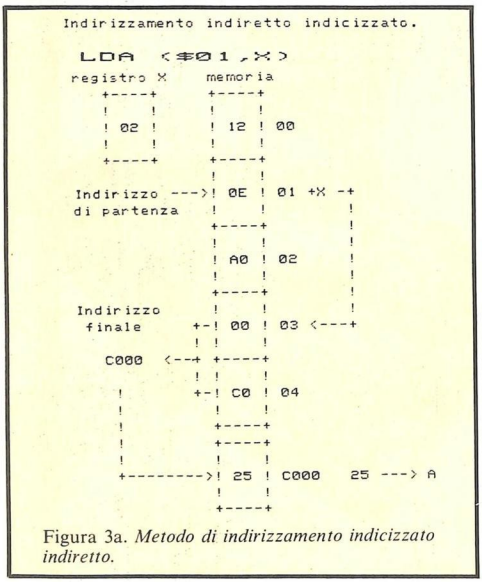


Figura 3a. *Metodo di indirizzamento indicizzato indiretto.*

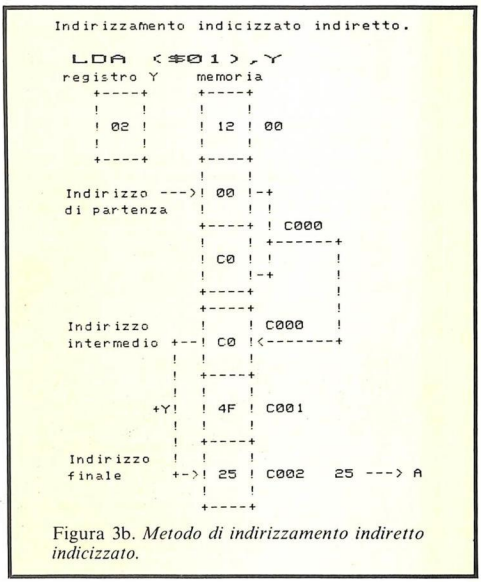


Figura 3b. *Metodo di indirizzamento indiretto indicizzato.*

dove la rapidità di esecuzione è essenziale.
Ad esempio, se volessimo scrivere una routine che esegua lo scroll dello schermo al contrario (verso il basso), non dovremmo fare altro che ricopiare il contenuto di ogni linea dello schermo sulla seguente, a partire dal basso verso l'alto.
Infine, la prima linea viene sostituita con una vuota. Questa operazione deve essere eseguita due volte, una per la mappa dello schermo e l'altra per la mappa dei colori. Perché, il tutto avvenga istantaneamente è indispensabile scrivere la routine in linguaggio macchina (listato 1).
In effetti, il lavoro si riduce a ricopiare due tabelle (caratteri e colori) quaranta locazioni di memoria dopo (equivalenti a una linea di schermo).
Nonostante le istruzioni utili si siano già viste tutte, avremmo notevoli difficoltà a scrivere tale routine con i metodi di indirizzamento che già conosciamo. Infatti, con l'indirizzamento assoluto, Y si può accedere a soli 256 elementi consecutivi mentre le due mappe video sono composte da 1.000 locazioni ciascuna.

Per risolvere questo problema, il 6502 è dotato di altri due tipi di indirizzamento: indicizzato-indiretto e indiretto-indicizzato. Entrambi utilizzano un registro come indice (come avveniva per l'assoluto, Y), ma l'indirizzo di base anziché essere assoluto viene letto in due locazioni consecutive in pagina zero. Per esempio, un indicizzato indiretto è:
LDA ($01,X)

Mentre è indiretto indicizzato:
LDA ($01),Y

Le figure 3a e 3b illustrano le differenze tra i due tipi di indirizzamenti. L'indicizzato indiretto prende l'indirizzo finale in due locazioni consecutive puntate dall'indirizzo in pagina zero specificato, più il registro X. In pratica permette di scegliere in una tabella di indirizzi.
Nell'indiretto indicizzato il 6502 si calcola l'indirizzo finale prendendo il contenuto della locazione in pagina zero specificata e della seguente, nell'ordine byte basso-byte alto, e vi aggiunge il contenuto del registro Y. L'indirizzamento ideale per la nostra routine di scroll è l'indiretto indicizzato. Si può sfruttare, infatti, il registro Y per ricopiare una linea ed aggiornare l'indirizzo contenuto nelle locazioni indirette per passare ad un'altra linea. Per evitare di perdere dei dati questa operazione va eseguita a partire dalla penultima linea.
Riferitevi al listato 1 che contiene i codici di tale routine.

Con ciò termina questo primo corso di linguaggio macchina per il C 64 e VIC 20. ■

Ecco come ti abboni, risparmi, vinci.

ETHOS

VINCI 100 COMMODORE 64

Abbonatevi subito: tra tutti coloro che si abboneranno a una o più riviste Jackson tra il 15/9/84 e il 28/2/85 saranno estratti a sorte mensilmente 20 Commodore 64.

Per sottoscrivere abbonamenti potrete utilizzare il modulo di cc/p inserito in questo fascicolo o inviare un assegno allegato al tagliando sottostante.

Gruppo Editoriale Jackson S.r.l. - Via Rossellini, 12 - 20124 Milano, allegando assegno o fotocopia della ricevuta di versamento con vaglia postale intestato a GRUPPO EDITORIALE JACKSON - MILANO.

☐ Si, desidero sottoscrivere un abbonamento a:

☐ Videogiochi (11 n.) L. 30.000 anzichè L. 38.500
☐ Home Computer (11 n.) L. 31.500 anzichè L. 38.500
☐ Personal Software (11 n.) L. 34.000 anzichè L. 44.000
☐ Bit (11 n.) L. 43.000 anzichè L. 55.000
☐ Informatica Oggi (11 n.) L. 31.000 anzichè L. 38.500
☐ PC Magazine (10 n.) L. 40.000 anzichè L. 50.000
☐ Personal O (10 n.) L. 35.000 anzichè L. 40.000
☐ Compuscuola (9 n.) L. 15.000 anzichè L. 18.000
☐ Telecomunicazioni Oggi (10 n.) L. 28.000 anzichè L. 35.000
☐ Automazione Oggi (11 n.) L. 30.500 anzichè L. 38.500
☐ Elettronica Oggi (11 n.) L. 36.000 anzichè L. 44.000
☐ L'Elettronica (22 n.) L. 44.000
☐ Elektor (12 n.) L. 29.000 anzichè L. 36.000
☐ Strumenti Musicali (10 n.) L. 24.000 anzichè L. 30.000

Attenzione per abbonamento all'estero le tariffe devono essere aumentate del 50%

E c'è un super-risparmio a chi si abbona a due o più riviste.

Tutti coloro che sottoscrivono l'abbonamento a due o più riviste godono di un prezzo ulteriormente agevolato, come appare nella seguente tabellina.

Esempio: Bit+Informatica Oggi L. 43.000+31.000=74.000 meno L. 2.000=L. 72.000

Abbonamento	
a 2 riviste L. 2.000 in meno sulla somma dei 2 prezzi d'abbonamento	a 8 riviste: L. 20.000 in meno sulla somma degli 8 prezzi d'abbonamento
a 3 riviste L. 4.000 in meno sulla somma dei 3 prezzi d'abbonamento	a 9 riviste L. 25.000 in meno sulla somma dei 9 prezzi d'abbonamento
a 4 riviste L. 7.000 in meno sulla somma dei 4 prezzi d'abbonamento	a 10 riviste L. 30.000 in meno sulla somma dei 10 prezzi d'abbonamento
a 5 riviste L. 10.000 in meno sulla somma dei 5 prezzi d'abbonamento	a 11 riviste L. 35.000 in meno sulla somma degli 11 prezzi d'abbonamento
a 6 riviste L. 13.000 in meno sulla somma dei 6 prezzi d'abbonamento	a 12 riviste L. 40.000 in meno sulla somma dei 12 prezzi d'abbonamento
a 7 riviste L. 16.000 in meno sulla somma dei 7 prezzi d'abbonamento	a 13 riviste L. 44.500 in meno sulla somma dei 13 prezzi d'abbonamento
	a 14 riviste L. 50.000 in meno sulla somma dei 14 prezzi d'abbonamento

☐ Allego assegno n° _____ della Banca _____
☐ Ho effettuato versamento con vaglia postale e allego fotocopia della ricevuta

Nome _____
Cognome _____
Azienda _____
CAP. _____ Città _____
Via _____

GRUPPO EDITORIALE JACKSON

COMPUTERWORLD

Perché il professionista del DP è cittadino del mondo

Ogni settimana, tempestivamente, notizie, anticipazioni, indiscrezioni raccolte all'origine dall'Italia e dal mondo.

COMPUTERWORLD ITALIA
è una pubblicazione

COMPUTER PUBLISHING GROUP Via Rosellini 12 20124 Milano

***Computerworld Italia** è il settimanale leader per il professionista del DP.*
*Viene inviato **esclusivamente** per abbonamento.*
Non perdere il contatto con il mondo: abbonati subito!

COMPUTERWORLD

Perché l'informatica è una professione

NEWS

SOFTWARE & SERVIZI

SISTEMI & PERIFERICHE

INDUSTRIA & MERCATO

OCCHIO AI MICRO

COMMUNICATIONS

SPECIALI

IN PROFONDITA'

Ogni settimana, regolarmente

Applicazioni *Organizzazione* *Mercato*
Metodologie *Prodotti* *Tecnologie*

SOLO IN ABBONAMENTO

Ritagliare e spedire in busta chiusa a:

Servizio Abbonamenti

Computer Publishing Group S.r.l. *Via Rosellini, 12 - 20124 MILANO*

- *Si, mi abbono per un anno al settimanale COMPUTERWORLD ITALIA (45 numeri - L. 78.000).*
- *Allego assegno di L. 78.000 n. _____ della Banca _____*
- *Ho effettuato versamento con cc/p n. 26732206 allego fotocopia ricevuta.*
- *Ho effettuato versamento tramite vaglia postale o telegrafico e allego fotocopia ricevuta.*

Cognome _____
Nome _____
Società _____
Indirizzo _____
C.A.P. _____ *Città* _____
(Prov. _____*)*

PERSONAL SOFTWARE

Dentro l'avventura

Parte seconda

Continua la costruzione del nostro Adventure per C 64

di *Roberto Tabacco*

Prima di riprendere il discorso interrotto la scorsa puntata penso sia bene chiarire e spiegare come si gioca un Adventure. Mi riferisco naturalmente solo agli Adventure in italiano, quei pochi che esistono in commercio.

C'é un terribile pregiudizio che scoraggia noi autori ed è quello solito che si attribuisce ad un prodotto ideato e realizzato in Italia. Ci sono Adventure americani o inglesi molto belli e ci sono adventure americani o inglesi molto belli e ci sono adventure americani o inglesi molto brutti. Come in tutte le cose anche in questo campo si possono trovare in Italia Adventure veramente belli, costruiti bene, con una storia interessante e coinvolgente. Per quanto mi riguarda cerco di costruire un Adventure pensando molto alla storia ed al suo svolgimento, creando sorprese, momenti particolari di gioco e perché no, tensione! Un Adventure non deve essere troppo difficile, ma neppure troppo facile; la storia non deve essere troppo originale, ma neppure troppo banale. I vari problemi debbono avere sì una difficoltà di soluzione, ma chiaramente trattandosi di un gioco che deve essere giocato, non debbono essere troppo astrusi o complicati.

Come autore di Adventure ho giocato molti adventure soprattutto americani. Conosco lo scoramento e la confusione che nasce appena il computer ci pone i primi interrogativi di gioco. Diciamo la verità non si sa assolutamente nulla, si è completamente all'oscuro di tutto. Il nostro primo impulso sarebbe di spegnere il computer e lasciar perdere.

L'avventura ci sembra troppo complicata e assolutamente ci rifiutiamo di entrare in quel particolare mondo.

Il problema è che non siamo abituati a giocare a questo genere di gioco, che invece ci può riservare delle autentiche emozioni. Vogliono scusarmi gli incalliti giocatori di Adventure se prima di andare avanti a spiegare come si realizza un adventure cerco di spiegare come si gioca un adventure. Come ogni gioco ci sono delle regole da seguire e bisogna conoscere la strategia. Il semplice gioco dell'oca se dato a chi mai l'ha giocato diventa complicato. Ora più o meno le regole di gioco sono sempre le stesse, la strategia invece cambia ad ogni nuovo gioco. Fare i cruciverba impone una regola di gioco, semplice, ma pur sempre una regola. Giocare agli Adventure impone un metodo logico e razionale di gioco. Quando accendete il computer ed il vostro gioco inizia a girare, vi viene presentata una schermata con delle indicazioni (più o meno precise). Il manuale, invece, vi da le premesse al gioco: non sottovalutatelo mai, leggetelo, è importante. Nel manuale oltre le indicazioni di come si gioca si possono trovare indizi che aiutano a giocare.

La prima volta che si gioca si commette l'errore di giocare credendo di arrivare subito dove si deve arrivare e non ci si cura di prendere delle piccole precauzioni. Ad esempo carta e matita. Si ignora la mappa e si comincia a giocare col computer e girare, sempre negli stessi luoghi. Non potete ricordare tutte le mosse che state facendo, è impossibile. Io stesso che ho ideato il programma quando lo mostro e voglio arrivare alla fine ho con me le indicazioni giuste scritte su un foglio. Quindi esaminate le stanze o i luoghi, annotate ciò che vi sembra importante, non smarritevi nel tortuoso labirinto di parole. Vedrete che con questo semplice metodo base riuscirete a percorrere il cammino senza girare a vuoto. Costruitevi insomma una specie di filo di Arianna. Quando mai si entra in un labirinto sconosciuto senza prendere i giusti attrezzi? Voi non siete il computer, non potete ricordare le mosse ed i luoghi passati dovete scriverli da qualche parte. Prendete la mappa tenetevela sempre sotto gli occhi e cominciate con una matita ad annotare i primi movimenti. Come si annotano i movimenti lo abbiamo già visto. Con il simbolo delle freccette. Un quadratino è il luogo di partenza e se andiamo a Nord segniamo una freccia verso Nord. Vediamo che succede a Nord ... e se c'é qualche pericolo, la prossima volta non passeremo a Nord. Vi sembra difficile? Se nella stanza che troviamo a Nord c'è qualche oggetto proviamo a prenderlo.

Potrà anche darsi che sia un oggetto da non prendere, ma potrà invece essere un oggetto utile. Se commettiamo un errore il gioco ci risponderà ci manderà forse a Game over, nessun problema. Ricominciamo da capo scrivendo che a Nord c'è un oggetto da non prendere. E così via! Generalmente nella schermata troviamo anche indicazioni che ci possono aiutare in seguito. Non tutti gli oggetti dobbiamo lasciarli: per alcuni dobbiamo usare verbi speciali e diversi da verbo lasciare. In un certo momento un oggetto ci serve per compiere una azione: ad esempio non dobbiamo dire lascio la pistola, ma Sparo con la pistola ... mi pare ovvio! - Altre volte possiamo

conoscere il verbo giusto se risolviamo un enigma. La nostra attenzione deve essere costante e non dobbiamo perdere una sola informazione. Certe volte ho sentito dire: Sono arrivato fino a quel punto eppoi non sono riuscito ad andare avanti. Vuol dire che qualcosa ti è sfuggito, che ti manca un oggetto o una parola chiave che è in qualche luogo che tu non hai ancora visitato. Se hai marcato tutto, ti accorgerai senza problemi di quel luogo rimasto ancora inesplorato. I giochi più sofisticati e completi inoltre vi danno la possibilità di salvare la posizione. Quando non riuscite ad andare avanti oppure siete stanchi salvate sempre la posizione. Quando riprenderete la sfida non sarete costretti a ripartire da capo. Oppure potete ripartire seguendo altre vie. Fate delle prove fino a che non trovate le giuste soluzioni.

Torniamo al nostro Adventure. Tutta una serie di comandi importanti risiedono nel modulo 1: i comandi dell'inventario, i comandi del prendere, lasciare ed esaminare gli oggetti del gioco. Lasciare-prendere ed esaminare, sono i verbi di uso comune nella nostra avventura. Abbiamo già visto che dovremo aggiungerne altri, ma ritornerò in seguito su questo argomento. A 2540 inizia l'inventario. Quando nel corso del gioco si prende un oggetto è giusto e logico che qualcosa nel programma stesso ci ricordi l'oggetto preso. La nostra memoria potrebbe ingannarci. Inventario nel gioco è insomma una sorta di utility. IV = é la variabile che indica appunto Inventario.

2542-Diciamo che IV è uguale a 0— e dopo facciamo riconoscere al computer che 5 sono gli oggetti che debbono essere raccolti. 5 sono gli oggetti nei DATA. Quindi per L seleziona da 1 a 5 e se OG%(L)=−1 allora stampa OG$(L).

Quindi aggiungiamo che IV=IV+1. OG%(L)=−1 vuol dire che l'oggetto è stato preso. Quindi chiudiamo il ciclo FOR con il NEXT. Rimandiamo tutto a 182, all'input D$. A 2550 c'è la routine del Prendere oggetti. È costruita più o meno come la precedente. La linea interessante è la 2553. SO$ è la variabile che indica il nome (che fa insomma riconoscere al computer che si tratta di un nome non di un verbo). Allora se OG$(G), il nome dell'oggetto nei DATA, è uguale al nome che poniamo noi all'input ... si salta a 2555 dove si incontra una istruzione che è un po' una novità: un piccolo, ma necessario algoritmo che ho ideato e che generalmente non si trova negli altri Adventure, e che esamineremo un'altra volta. Esaminiamo ora le risposte.

2556- Se OG%=−1 allora stampa-Hai già preso (nome oggetto) - a 2557 se OG%(G)<>PC allora stampa "Dove vedi" + nome oggetto. Vuol dire che se il numero che indica OG% non è uguale al numero di Stanza, PC significa che l'oggetto non è in quella stanza. Infine la 2558 dopo aver esaminato le precedenti dice che Hai raccolto + nome oggetto. A 2579 c'è invece la routine per il

Listato 1. *I comandi dell'inventario.*

```
2540 REM INVENTARIO
2541 PRINT"■HAI RACCOLTO :"
2542 IV=0:FORL=1TO5
2543 IFOG%(L)=-1THENPRINTOG$(L):IV=IV+1
2544 NEXT
2546 IFIV=0THENPRINT"NULLA E' NELLE TUE MANI!"
2547 GOTO182
2550 REM PRENDERE OGGETTI-
2552 FORG=1TO5
2553 IFOG$(G)=SO$THEN2555
2554 NEXT
2555 IFG=6THENGOTO2600
2556 IFOG%(■)=-1THENPRINT"HAI GIA' PRESO ";OG%(G):GOTO182
2557 IFOG%(G)<>PCTHENPRINT"DOVE VEDI ";OG$(G):GOTO182
2558 PRINT"  HAI RACCOLTO ";OG$(G):OG%(G)=-1
2559 GOTO182
2579 REM LASCIARE OGGETTI *
2580 FORL=1TO5
2581 IFOG$(L)=SO$THEN2584
2582 NEXT
2584 IFL=6THENPRINT"COSA VUOI LASCIARE?":GOTO182
2585 IFOG%(L)<>-1THENPRINT"NON HAI ";OG$(L):GOTO182
2586 PRINT"■HAI LASCIATO ";OG$(L):OG%(L)=PC
2587 GOTO182
2589 REM ESAMINO OGGETTI*
2590 :
2591 FORE=1TO5
2592 IFOG$(E)=SO$THEN2595
2593 NEXTE
2595 IFE=6THENGOTO2599
2596 IFOG%(E)=-1THENPRINT"L'HAI TRA LE MANI":GOTO9000
2597 IFOG%(E)<>PCTHENPRINT"DOVE VEDI "OG$(E)" ? ":GOTO182
2598 PRINT"STAI ESAMINANDO...";OG$(E):GOTO9000
2599 PRINT"ESAMINA DELL'ALTRO!":GOTO182
```

Listato 2. *Lista delle locazioni intestate.*

```
5000 Print"  GROTTA DELLO STREGONE   "
5001 Print"
5002 Print"
5003 Print"
5009 n=0:o=0:e=0:s=0:goto152
5010 Print"  DESERTO ALTO    "
5011 Print"
5012 Print"
5019 n=0:o=0:e=0:s=0:goto152
5020 Print"  DESERTO ALTO              "
5021 Print"
5022 Print"
5023 Print"
5029 n=0:e=0:s=0:o=0:goto152
5030 Print"  DESERTO ALTO     "
5031 Print"
5032 Print"
5033 Print"
5039 n=0:s=0:e=0:o=0:goto152
5040 Print"  GROTTA SIOUX      "
5041 Print"
5042 Print"
5049 n=0:o=0:e=0:s=0:goto152 :remfuori gioco
5050 Print"  DESERTO DI MEZZO-"
5051 Print"  Sabbie Mobili      "
5052 Print"
5059 n=0:e=0:s=0:o=0:goto6000:rem(fuori gioco)-
5060 Print"  CASA BOSCAIOLO     "
5061 Print"
5062 Print"
5069 n=2:s=12:e=8:o=6:goto152
5070 Print"  DESERTO DI MEZZO          "
5071 Print"
5072 Print"
5079 n=0:s=0:o=0:e=0:goto152
5080 Print"  DESERTO DI MEZZO "
5081 Print"
5082 Print"
5083 Print"
5084 Print"
5089 n=0:s=0:e=0:o=0:goto152
5090 Print"  DESERTO DI MEZZO
5091 Print"
5092 Print"
5099 n=0:e=0:o=0:s=0:goto152
5100 Print"  DESERTO BASSO
5101 Print"
5102 Print"
5103 Print"
5109 n=0:s=0:e=0:o=0:goto152
5110 Print"  DESERTO ALTO    "
5111 Print"
5112 Print"
5119 n=0:s=0:e=0:o=0:goto152
5120 Print"  DESERTO BASSO
5121 Print"
5122 Print"
5123 Print"
5129 n=0:s=0:e=0:o=0:goto152
5130 Print"  DESERTO BASSO    -"
5131 Print"
5132 Print"
5139 n=0:s=0:e=0:o=0:goto6000
5140 Print"  DESERTO BASSO    -"
5141 Print"
5142 Print"
5143 Print"
5144 Print"
5149 n=0:s=0:e=0:o=0:goto152
5150 rem******fine locazioni****
```

verbo lasciare. Penso non sia neppure il caso di spiegarla poiché è simile alla precedente. A 2591 inizia la routine di Esamino. Anch'essa è in tutto simile alla precedente: cambia solo la variabile che in questo caso è E.

Concludiamo questa parte con la lista delle locazioni intestate del modulo 2. Esse vanno da 5000 fino a 5249. Sono esattamente i nostri 15 luoghi d'azione del gioco. Per il momento non vi sono le indicazioni dei movimenti possibili, li aggiungeremo in un secondo tempo quando avremo deciso quali itinerari far percorrere al nostro eroe. L'unica che possiede le indicazioni di movimento è il luogo n. 7: la Casa del Boscaiolo, luogo da cui parte la nostra avventura.

La casa come potete vedere è completamente circondata dal Deserto. Deserto Alto quello a Nord, Deserto di Mezzo quello centrale nella mappa e Deserto Basso quello a Sud. Come potete vedere mancano anche le descrizioni di ogni luogo. Descrizioni che possono riempire, in questo caso, 8 righe. Ad esempio la 5000 è la descrizione base del luogo Grotta dello Stregone. Dalla 5001 alla 5008 è tutto libero, qui dobbiamo porre la nostra descrizione ed eventuali problemi da risolvere. Alla 5009 troviamo invece i comandi di direzione (per il momento tutti a zero). Come già detto in precedenza, questo è solo una tra i tanti modi per scrivere un Adventure. E vi accorgerete da soli come scrivere un Adventure sia stimolante soprattutto nel ricercare nuove soluzioni. Man mano che andremo avanti porremo delle varianti, delle soluzioni diverse a questo tipo di schema.

Ora che avete la Mappa, gli oggetti che sono stati posti, la descrizione dei luoghi, potete ad esempio per conto vostro arricchire questi luoghi di particolari descrizioni. Non potete certo limitarvi a presentare il luogo semplicemente sotto l'etichetta "Casa del boscaiolo", ma il vostro Adventure deve risultare ricco e preciso. ■

IN MILANO - VIA MASCHERONI, 14

IL VOSTRO "NUOVISSIMO" COMPUTER SHOP

Libri e riviste di elettronica e informatica.

CORSI CONTINUI TUTTO L'ANNO CON I MIGLIORI SOFTERISTI

IL TELEFONO È 02-437.385

COMMODORE VIC 20 E C64

I super BASIC

Parte seconda

di Alessandro Guida

Per chi avesse perso la prima parte di questo articolo, diciamo che si tratta di una carrellata tra i sistemi più diffusi per implementare nuove istruzioni BASIC.

I wedge

Wedge vuol dire cuneo ed è un sistema molto utilizzato per arricchire il vocabolario del 64 o del VIC 20 con nuovi comandi.
Il BASIC V 2.0 della Commodore ha una routine dell'interprete allocata nella pagina zero, in RAM. Questa routine, detta di Charget, ha il compito di leggere i codici o i caratteri delle linee BASIC, sia in modo diretto che in programma, saltando automaticamente gli spazi ed indicando per mezzo del carry se il carattere letto è un numero o no.
Grazie al fatto, che questa routine risiede in RAM, è possibile modificarla per inserirne una nostra che intercetti i nuovi comandi che non verrebbero interpretati dal BASIC normale. Vediamo la routine originale:

```
.. 0073 e6 7a       inc $7a     :Incrementa i registri $7A,7B
.. 0075 d0 02       bne $0079   : che puntano al carattere da
.. 0077 e6 7b       inc $7b     : leggere.
.. 0079 ad ** **    lda $****   :Carica il carattere
.. 007c c9 3a       cmp #$3a    :Se maggiore del '9 (hex 39)
.. 007e b0 0a       bcs $008a   : termina con il carry alto
.. 0080 c9 20       cmp #$20    :Se e' uno spazio passa al
.. 0082 f0 ef       beq $0073   : carattere successivo
.. 0084 38          sec         :Con questa operazione pone
.. 0085 e9 30       sbc #$30    : il carry a uno, senza alte-
.. 0087 38          sec         : rare l'accumulatore, solo
.. 0088 e9 d0       sbc #$d0    : se il codice e' <30 ('0)
.. 008a 60          rts         :Termina la routine con il
                                : codice nell'accumulatore
                                : e il carry alto se non si
                                : tratta di un numero.
```

A questo punto possiamo interporre una nostra routine che controlli se il carattere letto fa parte di uno dei nuovi comandi o no. Nel primo caso viene eseguito dalla nuova routine,nell'altro si torna alla vecchia Charget. Possiamo, per esempio, scrivere:

```
0073 4c 0d c0       jmp $c00d   :Salta alla nuova routine
0076 ea ea ea       nop         :
0079 ad ** **       lda $****   :Se non e' un nuovo comando
                                : torna qui.


                                :NUOVA ROUTINE
c00d e6 7a          inc $7a     :Incrementa i puntatori
c00f d0 02          bne $c013   :
c011 e6 7b          inc $7b     :
c013 8c 24 c0       sty $c024   :Salva i contenuti dei registri
c016 8e 22 c0       stx $c022   : X ed Y.
c019 a0 00          ldy #$00    :
c01b b1 7a          lda ($7a),y:Legge prossimo carattere
c01d c9 40          cmp #40     :Confronta con '@
c01f f0 08          beq $c028   :Se uguale passa ad eseguire il
                                : nuovo comando.
c021 a2 69          ldx #$69    :Altrimenti ristabilisce i
c023 a0 00          ldy #$00    : registri X e Y
c025 4c 79 00       jmp $0079   :Torna alla vecchia routine.
c028                            :INIZIO ROUTINE NUOVI COMANDI
```

Per semplificare il riconoscimento dei nuovi comandi si usa farli precedere da un carattere normalmente non utilizzato dal BASIC. In questo caso abbiamo scelto il simbolo "@".
Va notato, infine, che la Charget routine è chiamata in diversi punti del sistema operativo, mentre noi siamo interessati solo alla fase di esecuzione degli statement. Perciò è necessario controllare, oltre alla presenza del simbolo "@", se l'indirizzo di chiamata è $A7E6, che fa parte della routine Execute next statement. Tutte le altre chiamate devono essere ignorate.

Nel listato 1, in linguaggio macchina, abbiamo voluto aggiungere, a scopo dimostrativo, i comandi @L e @P. Questi vanno usati senza parametri e permettono di cancellare, sullo schermo, dal cursore a fine linea (@L) e dal cursore a fine pagina (@P).
Il sistema del wedge, senz'altro molto comodo da utilizzare, non è ancora l'ideale per chi necessita dell'aggiunta di un solo comando particolare. Anche perché ha bisogno di una fase di inizializzazione, per la modifica della Charget routine, e di una tabella di nuovi comandi che ne pregiudica la flessibilità.
Il sistema che vedremo ora risolve questo problema quando i comandi che si vogliono aggiungere sono a sé stanti.

Routine Stand-Alone

La soluzione molto semplice è di scrivere delle routine a sé stanti che svolgono determinati compiti, senza essere collegate a particolari caratteristiche del sistema operativo o dell'interprete BASIC. Queste routine possono essere eseguite con dei normalissimi SYS all'interno di un programma.
Ovviamente, per questa loro caratteristica, non permettono l'implementazione di strutture particolari tipo WHILE ... WEND, o ELSE ma risolvono egregiamente il loro compito quando si tratta di comandi tipo PLOT, SCROLL ecc.
La zona di memoria che si presta ottimamente a contenere queste routine è quella che va da $C000 a $CFFF nel 64. Il sistema per caricarvi i dati in l.m. è a scelta del programmatore. In generale se si tratta di brevi routine conviene incorporarle nel programma e trasferirle con un ciclo di FOR-READ-POKE-NEXT.
Altrimenti possono essere registrate a parte sul disco o su nastro e richiamate all'inizio del programma principale con una linea tipo:

```
10 IF C=0 THEN C=1:LOAD"CODE",8,1
```

Listato 1. *Listato Assembly della routine Wedge.*

```
c000 a9 4c      lda #$4c    :Inizializza la routine di CHARGET
c002 85 73      sta $73     :
c004 a9 0d      lda #$0d    :
c006 85 74      sta $74     :
c008 a9 c0      lda #$c0    :
c00a 85 75      sta $75     :
c00c 60         rts         :
c00d e6 7a      inc $7a     :INIZIO WEDGE
c00f d0 02      bne $c013   :Aggiorna puntatori
c011 e6 7b      inc $7b     :
c013 8c 26 c0   sty $c026   :Salva i registri X e Y
c016 8e 24 c0   stx $c024   :
c019 a0 00      ldy #$00    :
c01b ba         tsx         :Legge nello stack l'indirizzo
c01c bd 01 01   lda $0101,x : di chiamata della CHARGET ROUT.
c01f c9 e6      cmp #$e6    :Se byte basso <> E6 torna al
                            : alla CHARGET originale
c021 f0 07      beq $c02a   :
c023 a2 69      ldx #$69    :Ristabilisce X e Y
c025 a0 00      ldy #$00    :
c027 4c 79 00   jmp $0079   :Torna Charget
c02a bd 02 01   lda $0102,x :Se byte alto indirizzo <> A7
c02d c9 a7      cmp #$a7    : torna alla Charget originale
c02f d0 f2      bne $c023   :
c031 b1 7a      lda ($7a),y :Legge prossimo carattere
c033 c9 40      cmp #$40    :Se diverso '@' Charget orig.
c035 d0 ec      bne $c023   :
c037 c8         iny         :
c038 b1 7a      lda ($7a),y :Legge carattere successivo
c03a c9 50      cmp #$50    :Se uguale P va alla routine
c03c f0 0e      beq $c04c   : corrispondente
c03e c9 4c      cmp #$4c    :Se diverso L torna alla
c040 d0 e1      bne $c023   : Charget originale
                            :CANCELLA LINEA
c042 38         sec         :
c043 20 f0 ff   jsr $fff0   :Legge posizione cursore
c046 20 70 c0   jsr $c070   :Routine cancellazione da Y,X a
                            : fine linea.
c049 4c 60 c0   jmp $c060   :Termina
                            :CANCELLAZIONE FINE PAGINA
c04c 38         sec         :
c04d 20 f0 ff   jsr $fff0   :Legge posizione cursore
c050 20 70 c0   jsr $c070   :Cancella fine linea
c053 a0 00      ldy #$00    :Inizio linea successiva col.0
c055 e8         inx         :Incrementa numero linea
c056 e0 19      cpx #$19    :Se =25 fine
c058 f0 06      beq $c060   :
c05a 20 70 c0   jsr $c070   :Cancella linea
c05d 4c 55 c0   jmp $c055   :Ripete
c060 18         clc         :FINE WEDGE
c061 a9 02      lda #$02    :Aggiorna puntatori $7A,7B
c063 65 7a      adc $7a     : allo statement successivo
c065 85 7a      sta $7a     :
c067 a9 00      lda #$00    :
c069 65 7b      adc $7b     :
c06b 85 7b      sta $7b     :
c06d 4c 79 00   jmp $0079   :Salta alla vecchia CHARGET
                            :ROUTINE CANCELLAZIONE DA Y,X A
                            :  FINE LINEA
c070 a9 04      lda #$04    :Carica in $FB,$FC inizio
c072 85 fc      sta $fc     : mappa schermo.
c074 a9 00      lda #$00    :
c076 85 fb      sta $fb     :
c078 86 fd      stx $fd     :Salva registro X
c07a ca         dex         :Decrementa numero linea
c07b 30 10      bmi $c08d   :Se < 0 fine calcolo indirizzo
c07d 18         clc         :Somma #40 all'indirizzo
c07e a9 28      lda #$28    : inizio mappa schermo
c080 65 fb      adc $fb     :
c082 85 fb      sta $fb     :
c084 a9 00      lda #$00    :
c086 65 fc      adc $fc     :
c088 85 fc      sta $fc     :
c08a 4c 7a c0   jmp $c07a   :Ripete
c08d a9 20      lda #$20    :Codice spazio nell'accumulatore
c08f 91 fb      sta ($fb),y :Cancella locazione schermo
c091 c8         iny         :Passa alla successiva
c092 c0 28      cpy #$28    :Se Y=#40 allora fine
c094 d0 f9      bne $c08f   :Altrimenti ripete
c096 a6 fd      ldx $fd     :Ristabilisce X
c098 60         rts         :Termina
```

Listato 2. *Routine in Assembly per le istruzioni PLOT, SCROLL e COLOUR.*

```
                            :ROUTINE PLOT
c000 20 f1 b7   jsr $b7f1   :Controlla virgolo e legge primo
                            : parametro (X)
c003 e0 28      cpx #$28    :Se maggiore di 40 errore
c005 90 03      bcc $c00a   :
c007 4c 48 b2   jmp $b248   :Salta a ILLEGAL QUANT. ERROR
c00a 8a         txa         :Salva valore X
c00b 48         pha         :
c00c 20 f1 b7   jsr $b7f1   :Legge parametro successivo (Y)
c00f e0 19      cpx #$19    :Se maggiore di 25 errore
c011 b0 f4      bcs $c007   :
c013 68         pla         :Riprende X
c014 a8         tay         :
c015 20 f0 ff   jsr $fff0   :Posiziona cursore
c018 60         rts         :Torna al basic
                            :ROUTINE SCROLL
c019 20 f1 b7   jsr $b7f1   :Legge parametro
c01c 86 15      stx $15     :Lo salva in $15
c01e a0 07      ldy #$07    :
c020 ad 11 d0   lda $d011   :Selezione 24 linee ed azzera
c023 29 f0      and #$f0    : bit posizione
c025 85 14      sta $14     :Memorizza risultato in $14
c027 98         tya         :Mette valore bit posiz. in A
c028 05 14      ora $14     :Aggiunge nibble alto $D011
c02a 8d 11 d0   sta $d011   :Aggiorna $D011
c02d a9 00      lda #$00    :Ciclo di ritardo
c02f 85 a2      sta $a2     :
c031 a5 a2      lda $a2     :
c033 c5 15      cmp $15     :
c035 90 fa      bcc $c031   :
c037 88         dey         :Decrementa valore bit posizione
c038 10 e6      bpl $c020   :Se>=0 ripete
c03a 78         sei         :Disabilita interrupt
c03b 20 ea e8   jsr $e8ea   :Scroll di una linea in alto
c03e ad 11 d0   lda $d011   :Rimette a 7 i bit di posizione
c041 29 f0      and #$f0    :
c043 09 17      ora #$17    :
c045 8d 11 d0   sta $d011   :
c048 58         cli         :Riabilita interrupt
c049 60         rts         :Termina
                            :ROUTINE COLOUR
c04a 20 f1 b7   jsr $b7f1   :Legge modo colore
c04d e0 00      cpx #$00    :0=modo colore normale
c04f d0 09      bne $c05a   :
c051 ad 11 d0   lda $d011   :Disabilita colore esteso
c054 29 bf      and #$bf    :
c056 4c 63 c0   jmp $c063   :
c059 ea         nop         :
c05a e0 01      cpx #$01    :1=modo colore esteso
c05c d0 a9      bne $c007   :Se <> 1 errore
c05e ad 11 d0   lda $d011   :Abilita colore esteso
c061 09 40      ora #$40    :
c063 8d 11 d0   sta $d011   :
c066 20 f1 b7   jsr $b7f1   :Legge colore bordo
c069 8e 20 d0   stx $d020   :Aggiorna registro corrisp.
c06c 20 f1 b7   jsr $b7f1   :Legge colore sfondo 1
c06f 8e 21 d0   stx $d021   :Aggiorna registro corrisp.
c072 ad 11 d0   lda $d011   :Se modo colore normale
c075 29 40      and #$40    :  termina
c077 d0 01      bne $c07a   :
c079 60         rts         :
c07a 20 f1 b7   jsr $b7f1   :Legge colore sfondo 2,3,4 e
c07d 8e 22 d0   stx $d022   : aggiorna i registri corrisp.
c080 20 f1 b7   jsr $b7f1   :
c083 8e 23 d0   stx $d023   :
c086 20 f1 b7   jsr $b7f1   :
c089 8e 24 d0   stx $d024   :
c08c 60         rts         :Termina
```

Nel caso di più routine, e per dare un aspetto mnemonico alle nuove istruzioni, conviene assegnare a delle variabili, il cui nome richiami la funzione svolta, il valore dell'indirizzo di entrata delle routine. Per esempio:

```
20 PLOT=49152:SCROLL=49256
```

Bisogna solo prestare attenzione al fatto che nessun nome contenga al suo interno qualche comando del BASIC. Ad esempio la variabile Color darebbe errore poiché contiene la funzione OR. Basta comunque modificarla in Colour.

A questo punto è sufficiente richiamare la funzione con un SYS seguito dal nome del comando che si vuole eseguire, ad esempio:

```
SYS SCROLL.
```

Anie / Ente autonomo per le Fiere di Bologna / Fondazione G. Marconi

SIOA '85 SALONE DELL' INFORMATICA, TELEMATICA, ORGANIZZAZIONE AZIENDALE

Bologna 16-20 febbraio 1985
Quartiere fieristico

dimensione Uomo

Consorzio Sioa - Via Napoli 20 Bologna - Tel. 051/452936 466911 - Tx 510878

Listato 3. *Dimostrativo utilizzante le routine del listato 2.*

```
5 print"■":gosub 50000:rem lettura routine lm
7 rem **** entry point delle routine ***
10 plot=49152:scroll=49177:colour=49226
20 print"■":sys scroll,15:rem *** attiva schermo 24 righe
25 sys colour,1,6,6,0,3,2:rem *** seleziona modo colore esteso ed i colori
30 for x=0 to 3
40 read a$: rem *** legge la stringa e la stampa
50 sys plot,0,24:print a$;:rem *** nella linea invisibile
60 sys scroll,15:rem *** esegue lo scroll
70 next:end
80 data "prova di scroll continuato"
90 data "CON  VIDEO IN  MODO COLORE":rem battere con shift (spazi compresi)
100 data"■esteso.  i colori sono blu"
110 data"■NERO BIANCO E ROSSO┐      ":rem battere con shift (spazi compresi)
50000 restore:in=49152
50005 read a$,b$,c$,d$
50010 reada$:ifa$="***"thenrestore:return
50020 tt=val(a$):pokein,tt
50030 forc=1to7:reada:pokein+c,a:tt=tt+a:next
50040 reada:ifa<>(tt and255)thenprint"errore nei data alla linea"peek(64)*256+pe
ek(63):end
50050 in=in+8:goto50010
50060 data032,241,183,224,040,144,003,076,175
50070 data072,178,138,072,032,241,183,224,116
50080 data025,176,244,104,168,032,240,255,220
50090 data096,032,241,183,134,021,160,007,106
50100 data173,017,208,041,240,133,020,152,216
50110 data005,020,141,017,208,169,000,133,181
50120 data162,165,162,197,021,144,250,136,213
50130 data016,230,120,032,234,232,173,017,030
50140 data208,041,240,009,023,141,017,208,119
50150 data088,096,032,241,183,224,000,208,048
50160 data009,173,017,208,041,191,076,099,046
50170 data192,234,224,001,208,169,173,017,194
50180 data208,009,064,141,017,208,032,241,152
50190 data183,142,032,208,032,241,183,142,139
50200 data033,208,173,017,208,041,064,208,184
50210 data001,096,032,241,183,142,034,208,169
50220 data032,241,183,142,035,208,032,241,090
50230 data183,142,036,208,096,255,000,000,152
50240 data ***
```

Chi ha paura di Jackson?

Con una incredibile decisione Mondadori rifiuta pubblicità su *Panorama*. Il cliente va all'Espresso.

Finché ci capita di leggere che nel Triveneto il Gazzettino si rifiuta di pubblicare un necrologio «in morte di un amico gay», ci arrabbiamo parecchio ma non ci stupiamo affatto; quando però una birbonata porta un cognome come Mondadori, il fatto ci stupisce assai e ci inquieta anche un poco. È successo che Panorama ha negato lo spazio pubblicitario alla campagna abbonamenti delle testate, tutte specializzate nell'area dell'informatica e dell'elettronica, del gruppo editoriale Jackson, perché - questa la motivazione addotta - «in concorrenza con la linea editoriale della Mondadori».
(Com'è noto nel pacchetto del gruppo di Segrate, tra le riviste che trattano di computer, spicca Zerouno, direttore Italo Cammarata). «Lo trovo un atteggiamento assurdo e ridicolo, purtroppo però anche molto pericoloso da un punto di vista più generale, per la libera circolazione in un libero mercato di idee e di prodotti», ha dichiarato a *Pubblico* Alberto Vitali, amministratore delegato della Ethos (l'agenzia ha acquisito l'editoriale Jackson proprio quest'anno). Per noi, prosegue Vitali, l'editoriale Jackson, non è un problema, dirotteremo l'investimento su *Repubblica* e *Espresso*. Al di là di tutto è soprattutto il modo in cui è stato gestito l'episodio che offende Vitali: Segrate ha rispedito gli esecutivi un giorno prima dell'uscita della campagna, accompagnandoli con una spiegazione molto laconica e telefonica da parte di una segretaria della direzione pubblica. Ethos ha comunque chiesto un parere ed un chiarimento in materia all'avvocato Fusi.

(dalla newsletter settimanale "Pubblico" nº 212 - 26 ottobre 1984).

I SEGRETI DEI PERSONAL

Generalmente è necessario trasmettere alla routine in linguaggio macchina, anche dei parametri. Questi possono essere messi direttamente dopo il nome della variabile, sarà la nostra routine che si preoccuperà di andarli a leggere.
Infatti, il comando SYS cede il controllo alla routine in linguaggio macchina prima di controllare se dopo l'indirizzo vi sono i due punti o la fine della linea. Perciò possiamo leggere i caratteri seguenti la variabile-indirizzo con la routine di Charget. Possiamo, anche, servirci di routine, tipo $B7F1 che controlla se vi è una virgola e trasferisce il dato seguente la virgola nel registro X, gestendo gli eventuali messaggi di errore.
Se, per esempio, si vuole indirizzare il cursore in un punto qualsiasi dello schermo, potremmo dare in BASIC il comando:

50 SYS PLOT, 10, 20

ed avere la corrispondente routine in linguaggio macchina:

```
c003 e0 28      cpx #$28    :Se >40
c005 90 03      bcc $c00a   :
c007 4c 48 b2   jmp $b248   :   salta ad ILLEGAL QUANT.ERR.
c00a 8a         txa         :
c00b 48         pha         :Conserva il primo parametro
c00c 20 f1 b7   jsr $b7f1   :Controlla seconda virgola e legge
                            : e preleva secondo parametro
c00f e0 19      cpx #$19    :Se =>25
c011 b0 f4      bcs $c007   :   ILLEGAL QUANT.ERR.
c013 68         pla         :Mette in Y primo parametro
c014 a8         tay         :
c015 20 f0 ff   jsr $fff0   :Subroutine spostamento cursore
c018 60         rts         :Termine
```

Al termine della routine i puntatori $7A e $7B della routine di Charget devono essere posizionati sullo # 00 di fine linea o sul ":" che separa due comandi, altrimenti al ritorno in BASIC si ha un SYNTAX ERROR. Nell'esempio precedente, la routine $B7F1 si occupa anche di prelevare il carattere seguente.
Vi sono molte altre routine utili che permettono di prelevare numeri e stringhe da una linea BASIC. Chi è interessato può trovarne un elenco completo nel volume *Il sistema operativo del C 64.*
Riportiamo un esempio di applicazione in cui sono stati implementati i comandi PLOT, SCROLL e COLOUR. Naturalmente, è possibile utilizzare anche solo una di queste routine.
La sintassi dei comandi è:

SYS PLOT,x,y - Sposta il cursore in un punto qualsiasi dello schermo (x,y coordinate schermo);

SYS SCROLL,v - Esegue lo scroll lento (un pixel per volta) dello schermo (v è la velocità dello scroll: 0-max);

SYS COLOUR, t, b, s1 (,s2,s3,s4) - Controlla i colori dello schermo.

(t=0-modo normale/t=1-modo colore sfondo esteso);
(b=colore bordo);
(s1=colore sfondo);
(s2,s3,s4=colori alternativi sfondo necessari se t=1).

Il modo colore di sfondo esteso permette di scrivere con diversi colori come sfondo, rinunciando al set di caratteri che si ottiene con lo shift e a quello in reverse. In pratica un PRINT normale stampa nel colore s1, un PRINT di caratteri con lo SHIFT stampa con colore di sfondo s2, il reverse attiva il colore s3 ed infine il reverse + lo SHIFT il colore dello sfondo s4.
I listati 2 e 3 dovrebbero risolvere ogni dubbio.

SHARP PC-1251

Alla scoperta del linguaggio macchina

di Mauro Lenzi

Prima di addentrarci nell'interessante argomento di questo mese, desidero porre alla vostra attenzione due notizie che io giudico particolarmente importanti.
La prima riguarda la nascita di un club degli "sharpisti", gestito dalla concessionaria per l'Italia della Sharp, la ditta Melchioni, e con iscrizione completamente gratuita. L'importanza di queste iniziative deve essere giustamente apprezzata: quando si acquista un personal computer, anche se tascabile, non è sufficiente valutarne le caratteristiche tecniche. Infatti bisogna dare un peso determinante anche a due altri fattori: la diffusione e il software reperibile in commercio.
Acquistare un computer molto conosciuto ed avere contatti con altre persone che lo possiedono per scambiarsi vicendevolmente programmi e "segreti" è una cosa importantissima ed ora, finalmente, anche gli sharpisti potranno farlo.
La seconda notizia, per tutti quei lettori che hanno avuto la pazienza di seguirmi fino ad ora verso la scoperta dei segreti dello Sharp PC-1251 è addirittura sensazionale: presto saranno disponibili i manuali tecnici di questo computer e dello Sharp PC-1500! Sono infatti già comparsi nei cataloghi e prestissimo dovrebbero essere a disposizione di tutti. Unico piccolo difettuccio è che sono in inglese, ma, a quanto dice il catalogo, dovrebbero contenere tutte le spiegazioni necessarie alla programmazione in linguaggio macchina e gli schemi elettrici dei sistemi. Forse finalmente saranno svelati alcuni misteri che ci hanno fatto passare lunghe notti insonni!

A questo punto è uno sforzo inutile tentare di decifrare il linguaggio macchina del nostro Sharp con i metodi artigianali che abbiamo usato fino ad ora, perché l'im-

Alla scoperta del linguaggio macchina

presa richiederebbe uno sforzo immane e tantissimo tempo, tuttavia questi articoli vogliono illustrare dei metodi generali per giungere ad avere una conoscenza veramente approfondita del proprio personal computer. Decifrare il linguaggio macchina significa sapere tutto del nostro computer, ma è un'impresa estremamente difficile.

Quello che adesso darò è semplicemente uno spunto, frutto di semplici considerazioni, che sicuramente potrebbe essere migliorato e può portare anche a conclusioni non sempre esatte e che richiederebbero ulteriori conferme a posteriori. Naturalmente dobbiamo supporre che il linguaggio macchina dello Sharp PC-1251 sia simile a quello di tutti gli altri computer, anche se abbiamo visto che in molte occasioni esso presentava caratteristiche singolari.

La prima cosa che dobbiamo cercare è un'istruzione di salto, una specie di GOTO o di GOSUB del linguaggio BASIC.

Il metodo per trovare le istruzioni di questo tipo è molto semplice, con le conoscenze che abbiamo; supponiamo di volere richiamare la subroutine in linguaggio macchina corrispondente al comando BASIC Radian. Negli articoli precedenti si è visto come è possibile richiamarla dal BASIC: sappiamo che questa subroutine incomincia all'indirizzo di memoria 88 x 256 + 140 = 22668, perciò basterà digitare CALL 22668.

Un comando in linguaggio macchina è costituito naturalmente da un numero, compreso fra zero e 255, cioè occupante 1 byte. Alcuni comandi, però, possono richiedere che vengano specificati alcuni dati, che occuperanno altri byte. Nel nostro caso, cioè, per trovare un'istruzione di salto, dovremo specificare anche l'indirizzo di memoria da cui inizia la subroutine. Poichè questo indirizzo è maggiore di 255, dobbiamo dividerlo in "due pezzi": questi pezzi li conosciamo già e sono i numeri 88 e 140. Concludendo, dunque, possiamo supporre che un'istruzione di salto in linguaggio macchina sarà della forma X 88 140, dove X è un numero che dobbiamo trovare e corrisponderà ad un comando simile a GOTO o a GOSUB, mentre 88 140 rappresentano l'indirizzo iniziale della subroutine Radian.

Si tratta ora di determinare il numero X. Supponiamo che sia uguale a zero. In tal caso se immettiamo in qualche parte della memoria RAM, ad esempio a partire dall'indirizzo 40000, la sequenza 0 88 140 e poi facciamo eseguire questa istruzione digitando CALL 40000, vedremo che il computer si predisporrà ad eseguire i calcoli in radianti.

Se invece 0 non corrisponde ad un comando di salto in linguaggio macchina del tipo di quelli che stiamo cercando, eseguendo CALL 40000 si possono verificare due casi: o non succede assolutamente niente oppure il computer "muore", ed in tal caso non ci rimane che usare il tasto di Reset per "resuscitarlo".

4 5 6 7 8 10 11 12 13 14 15 20 21 24 26 28 29 30 32 33 34 35 36 37 38 39 48 49 51 53 64 65 66 67 68 69 70 71 72 73 74 77 80 81 82 83 84 85 86 87 88 89 90 93 95 104 106 122 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 194 195 196 197 198 199 200 201 202 203 204 205 206 210 211 217 218 219 221

Figura 1. *I numeri rappresentati corrispondono molto probabilmente ad istruzioni in linguaggio macchina ad un solo byte.*

Con un ragionamento analogo è possibile, sostituendo ad X tutti i numeri da 0 a 255, trovare tutte le istruzioni di salto del linguaggio macchina.

La cosa sembra più breve a dirsi che a farsi, infatti richiede una pazienza certosina, dovendo eseguire 256 tentativi e tantissimi Reset!!

In realtà con questo metodo non troveremo proprio tutte le istruzioni di salto, perché, come nel BASIC, anche nel linguaggio macchina esistono dei salti condizionati, simili agli IF THEN, che non sapendo a quali condizioni debbono soddisfare, saranno eseguiti oppure no senza alcun controllo da parte nostra. Nel linguaggio macchina esistono poi anche dei salti relativi, che corrisponderebbero ad un'ipotetica istruzione BASIC che faccia saltare avanti o indietro di un certo numero di linee di programmazione.

Col procedimento descritto sono riuscito a stabilire che i numeri 121 124 e 125 dovrebbero corrispondere a comandi di linguaggio macchina del tipo cercato: così inserendo all'indirizzo 40000 la sequenza 121 88 140 e poi richiamandola col comando CALL 40000 il computer si predisporrà a eseguire i calcoli in radianti.

Un ulteriore passo avanti che possiamo fare nella conoscenza del linguaggio macchina sconosciuto è di determinare la lunghezza dei vari comandi: comandi come quelli che abbiamo cercato, occupano complessivamen-

Listato 1. *Questo semplice programma permette di controllare se un certo numero N corrisponde ad una istruzione in linguaggio macchina ad un solo byte: in caso affermativo, quando il programma eseguirà la linea 40 il computer si predisporrà a fare i conti in radianti e si udirà un beep. Modificando la linea 10 come segue:*
10 X = 40000 : POKE X+1, 55, 124, 88, 140
è possibile verificare le istruzioni a due byte.

```
10  X= 40000: POKE X+1, 124, 88, 140
20  INPUT N
30  DEGREE
40  POKE X, N : CALL X
50  IF SIN π = 0 THEN BEEP 1 : GOTO 20
60  END
```

te tre byte, uno per specificare il comando vero e proprio e due per specificare un certo indirizzo di memoria. Altre istruzioni possono richiedere solo due byte: sono di questo tipo generalmente le istruzioni di salto relativo nelle quali il primo byte è occupato dal comando ed il secondo dice quanti byte si debbono saltare in avanti o all'indietro (si ha perciò che il salto non potrà essere di più di 256 byte). Infine si hanno delle istruzioni di un solo byte, come ad esempio l'istruzione corrispondente al comando BASIC, RETURN. Tutti i microprocessori più semplici non hanno istruzioni lunghe più di tre byte. È chiaro quindi che il prossimo passo consisterà nel cercare di stabilire quali istruzioni sono ad un byte, quali a due e quali a tre.
Il metodo che si può utilizzare a questo scopo è molto simile al precedente: consideriamo la sequenza X 124 88 140. Possono verificarsi due possibilità: se il numero X corrisponde ad una istruzione ad un byte, il computer la eseguirà e poi andrà ad eseguire quella successiva di tre byte che sappiamo già farà predisporre il computer al calcolo in radianti; se invece X è il numero di un comando a due o tre byte, interpreterà il numero 124 non come un successivo comando, ma come un dato, e perciò il computer invece di predisporsi in radianti farà "cose strane". Sostituendo al numero X tutti i numeri da 0 a 255 possiamo distinguere tutti quelli corrispondenti ad istruzioni ad un byte. In figura 1 sono trascritti tutti i numeri che, molto probabilmente corrispondono ad istruzioni ad un byte. Il listato 1 è invece un semplice programma di verifica: immettendo in N un numero corrispondente ad una istruzione in linguaggio macchina ad un byte si udirà un Beep.

Nello stesso modo, immettendo in memoria la sequenza X 55 124 88 140 è possibile scoprire alcune istruzioni sicuramente a due byte. Il numero 55 è stato posto a caso, essendo semplicemente un dato, ma probabilmente sostituendolo con altri numeri potremmo scoprire altre istruzioni a due byte. Istruzioni sicuramente a due byte sono quelle rappresentate in figura 2. Con ragionamenti di questo tipo e con tantissima pazienza sarebbe possibile progredire a piccoli passi fino a raggiungere una completa conoscenza del linguaggio macchina del nostro computer: tuttavia, come detto all'inizio, ho voluto solo dare un cenno di come si potrebbe procedere in questa ricerca, perché presto avremo a disposizione i nuovi manuali coi quali, spero, potremo fare un grosso passo avanti nella conoscenza del PC-1251.

·0 1 2 3 17 18 19 56 57 58 59 78 96 97 98 99 101 103 112 113 114 115 116 117 118 119 212 213 214 225 233 234 237 242 244 245 247 251 254

Figura 2. *Sono rappresentati in questa figura i numeri corrispondenti ad istruzioni a due byte. Sicuramente questi sono solo una parte, perché molte istruzioni a due byte servono ad eseguire salti "relativi" e per scovarle bisognerebbe fare altre semplici prove.*

SINCLAIR ZX SPECTRUM

L'uso e ... l'abuso dei caratteri di controllo

di Marcello Spero

Questo mese presentiamo un intervento del lettore Tullio Policastro, su un argomento di sicuro interesse. Probabilmente la maggior parte degli utenti di quell'ottimo computer che è il nostro ZX Spectrum, anche se abbastanza esperti di programmazione, quando sono incappati in quella parte del Capitolo 16 del Manuale d'uso (in inglese) che tratta brevemente l'argomento dei caratteri di controllo l'avranno considerato irrilevante per i propri scopi, poco comprensibile, e l'avranno saltato a piè pari.
In effetti, per la maggior parte delle applicazioni correnti l'uso dei caratteri di controllo - ossia di quella sezione del set di caratteri non-standard dello Spectrum che va dal codice 0 al codice 31, risulta inutile, o meglio può essere sostituito da altri comandi di più semplice ed immediata comprensione. È il caso per esempio delle istruzioni di PRINT, per le quali sono previsti, oltre a tre diversi comandi di formattazione (stampa consecutiva, a capo riga o all'inizio di una mezza riga), ben 6 comandi per gli attributi (INK, PAPER, OVER, INVERSE, FLASH, BRIGHT); nonchè un posizionamento fine mediante i comandi TAB ed AT (quasi tutti applicabili pure all'istruzione INPUT).

Ciò non toglie che la conoscenza non superficiale delle possibilità di utilizzazione dei caratteri di controllo possa offrire al programmatore più smaliziato l'opportunità di introdurre degli effetti utili.
Precisiamo innanzitutto che dei 32 caratteri di controllo citati, un certo numero sono già noti ed utilizzati: corrispondono ai codici da 4 a 15, e sono normalmente messi in funzione premendo Caps SHIFT ed i tasti numerici da 1 a 0 (corrispondono ai comandi scritti in bianco sopra questi tasti, da EDIT a DELETE), ai quali si aggiungono il tasto ENTER (carattere di controllo codice 13) e il passaggio in modo "E" premendo ambedue i tasti SHIFT insieme (carattere di controllo 14).
I rimanenti caratteri di controllo si riferiscono soprattutto ai comandi per gli attributi. Da un loro uso intelligente si possono ottenere effetti interessanti che passeremo ad esaminare un po' più avanti, dopo aver ricordato che ad essi si aggiungono pure 4 controlli di formatta-

L'uso e... l'abuso dei caratteri di controllo

zione di stampa; prima però mostreremo come si possono mettere in opera i vari caratteri di controllo.
La lista di quelli che ci interessano si è dunque ridotta a questi:

Codice	Funzione usuale	Funzione in stampa
0	FLASH 0	--
1	FLASH 1	--
2	BRIGHT 0	--
3	BRIGHT 1	--
6	CAPS LOCK	a capo mezza riga
8	freccia a sinistra	torna indietro di un carattere
13	ENTER	a capo nuova riga
16	PAPER nero (0)	controllo INK
17	PAPER blu (1)	controllo PAPER
18	PAPER rosso (2)	controllo FLASH
19	PAPER viola (3)	controllo BRIGHT
20	PAPER verde (4)	controllo INVERSE
21	PAPER azzurro (5)	controllo OVER
22	PAPER giallo (6)	controllo AT
23	PAPER bianco (7)	controllo TAB
24	INK nero (0)	--
25	INK blu (1)	--
26	INK rosso (2)	--
27	INK viola (3)	--
28	INK verde (4)	--
29	INK azzurro (5)	--
30	INK giallo (6)	--
31	INK bianco (7)	--

Come si vede, la funzione - ed anche il modo di messa in opera - dei caratteri di controllo è diversa a seconda che essi vengano inseriti in un'istruzione (in modo immediato o in un programma), o che siano utilizzati in un comando di stampa o equivalente (PRINT, INPUT ed anche - inaspettatamente - SAVE).

Per ottenere la funzione normale si adopera un modo leggermente elaborato, che occorre seguire con attenzione per evitare strane sorprese.
Fortunatamente il Manuale d'uso riporta il modo di operare in una tabellina verso la fine del Capitolo 16, a cui si potrà ricorrere in caso di dubbio. Prima di tutto occorre passare in modo "E" (esteso), il che si ottiene come sappiamo premendo contemporaneamente i due tasti Caps SHIFT e Symbol SHIFT (compare appunto il cursore E lampeggiante).
A questo punto basterà premere - con o senza Caps SHIFT - uno dei tasti numerici per ottenere il carattere di controllo desiderato, e cioè uno di quelli che controllano, in modo immediato, INK, PAPER, FLASH e BRIGHT. Per i primi due attributi non occorre particolare memoria: se è premuto Caps SHIFT assieme al tasto numerico che si riferisce ad un codice di colore (da 0 a 7) si ottiene INK di quel colore; senza Caps SHIFT si ottiene PAPER di quel colore (ricordate che il colore è scritto in ... colore sopra il tasto numerico corrispondente). Inoltre, Caps SHIFT con un tasto 8 o 9 corrispondono a FLASH 0 e FLASH 1, mentre senza Caps SHIFT si hanno BRIGHT 0 e BRIGHT 1. È ovvio che, come con i comandi diretti, si possono inserire tramite i caratteri di controllo anche più attributi in successione: occorre però precisare che bisogna ogni volta ritornare preventivamente in modo E con la pressione dei due SHIFT.

Ora che si è spiegato come ottenere un dato carattere di controllo degli attributi, vediamo dove possiamo inserirli. Innanzitutto all'interno di una stringa qualsiasi (esplicita), ossia in una posizione qualunque fra le virgolette ("...") che la delimitano, proprio come un carattere normale, con la differenza che, pure esplicando il suo effetto su tutta la parte della stringa che lo segue, questo tipo di carattere sarà invisibile. È abbastanza ovvio, comunque, che la posizione più naturale sia all'inizio della stringa, subito dopo le prime virgolette: va precisato però che con i comandi diretti PAPER, INK ecc., l'effetto è temporaneo, ossia dura solo fino al termine di una istruzione o di uno statement entro un'istruzione. La caratteristica insolita che si osserverà sarà che il suo effetto è immediato, ossia si esplica già sui caratteri che si impostano subito dopo nella scrittura d'un comando diretto o di un'istruzione.

Questa caratteristica torna comoda quando si vogliano evidenziare in modo opportuno le varie parti di un listato, o determinate parti entro un'istruzione. Infatti, oltre all'inserimento entro una stringa, uno o più caratteri di controllo si possono inserire immediatamente dopo un numero di istruzione. Ma attenzione: in questo caso l'effetto non si manifesterà in fase di editing delle istruzioni successive, ma continuerà tuttavia ad esercitarsi sul listato! Quando perciò si vuole conservare l'effetto su di una determinata parte del listato e non oltre, occorre inserire un nuovo carattere di controllo (o più caratteri di controllo) che stabiliscano nuovi attributi (ad esempio ma non necessariamente, riportino a PAPER 7/INK 0 ecc.) al termine dell'ultima istruzione, prima di premere ENTER. Da notare che l'inserimento dei caratteri di controllo nelle istruzioni può avvenire comunque in qualunque posizione della linea, ed anche in fase di editing!

Per rendere queste spiegazioni un poco più chiare, valga un esempio.
Supponiamo di avere in memoria un programma di un certo numero di istruzioni, e di volerlo in un secondo tempo arricchire con delle REM che separano e brevemente illustrano le varie sezioni del programma, che si vogliono mettere in evidenza usando opportunamente certi attributi. Ammettendo che le istruzioni procedano di 10 in 10, inseriremo perciò una linea 5 REM, operando in questo modo (supponendo che la prima sezione che parte da questa REM debba avere PAPER giallo ed INK blu):

L'uso e... l'abuso dei caratteri di controllo

- impostare 5;
- premere i due SHIFT assieme (passaggio in modo "E");
- premere tasto 6 (PAPER giallo);
- ripremere assieme i due SHIFT;
- Caps SHIFT + tasto 1 (INK blu);
- impostare REM seguito dal relativo breve testo esplicativo;
- terminare con ENTER.

Si osserverà che *tutto* il programma a partire dal termine REM della linea 5 è passato a PAPER giallo ed INK blu! Ammettiamo ora di voler inserire prima della linea 100 un'altra REM, e che la successiva sezione debba risultare in PAPER blu ed INK bianco, lampeggiante. Si opererà così:

- impostare 95 (n° di riga che precede il 100);
- premere i due SHIFT assieme (modo E);
- premere tasto 7 (INK bianco);
- ripremere i due SHIFT;
- premere Caps SHIFT + tasto 1 (PAPER blu);
- ripremere i due SHIFT ...;
- premere Caps SHIFT + tasto 9 (FLASH 1);;
- impostare REM più testo relativo;
- terminare con ENTER.

Ancora una volta, tutto il programma a partire dalla nuova REM subirà l'effetto dei nuovi attributi.
Mostriamo ora come si può ad un certo punto annullare, a partire da una certa linea del programma, l'effetto degli ultimi caratteri di controllo inseriti. Ammettendo che dalla istruzione 300 si voglia tornare ad un listato normale (nero su bianco), si posizionerà il cursore di editing sulla linea 290 immediatamente precedente (ad esempio con LIST 290) e si richiamerà la linea con EDIT (Caps SHIFT + 1). Con l'uso dei tasti delle "frecce" (5 ed 8) ci si porterà a fondo linea.
A questo punto:

- premere i due SHIFT assieme (modo E);
- premere tasto 7 (PAPER bianco);
- ripremere i due SHIFT;
- premere Caps SHIFT + tasto 0 (INK nero);
- ripremere i due SHIFT (nell'ipotesi che fosse in atto FLASH 1);
- premere Caps SHIFT + tasto 8 (FLASH 0)(c.s.);
- terminare con ENTER.

Il resto del listato a partire dalla linea 300 sarà tornato normale.
È evidente che, tenendo ben conto dei modi per l'inserimento dei caratteri di controllo sopra citati, è possibile modificare a piacimento le varie parti di un programma, sia agendo su intere sezioni, che su parti di un'istruzione, sia entro le stringhe (fra le ""), assegnando gli attributi che si desiderano. Nel caso delle stringhe essi, oltre a risaltare nel listato, agiranno pure in fase di esecuzione del programma, con la stampa della stringa con quegli stessi attributi.

Un caso molto particolare di applicazione dei concetti ora esposti consiste nel mascheramento di una parte o addirittura di tutto il programma, salvo per il 1° numero di linea .. Sarà sufficiente usare per INK e PAPER inseriti con caratteri di controllo valori (codici) eguali, rendere invisibile il listato (almeno sino a quando, editata nella parte inferiore dello schermo la prima linea del programma, non si proceda a cancellare con DELETE (Caps SHIFT + 0) gli invisibili caratteri di controllo posti dopo il numero di linea, concludendo naturalmente con il reinserimento della linea con ENTER.
Ci resta da illustrare il modo di inserimento e di applicazione dei caratteri di controllo relativi alla seconda parte della nostra tabellina (terza colonna). Innanzitutto occorre chiarire che qui l'inserimento, con qualche eccezione che vedremo, può avvenire solo tramite la funzione CHR$ seguita dal numero di codice del particolare carattere di controllo. Con l'eccezione però del solo codice 8, che, come illustra a sufficienza il Manuale d'uso del solito Capitolo 16, può servire a sovrapporre due caratteri previa retrocessione di uno spazio operata appunto dal CHR$8, gli altri caratteri di controllo trovano scarsa utilità se impiegati assieme a PRINT (od INPUT), perché sostituibili più direttamente e chiaramente con i comandi diretti degli attributi o di posizionamento (TAB, AT, virgola, apice). Vi sono tuttavia alcune applicazioni particolari e poco note che merita ricordare.

Quando in un programma capiti di dover frequentemente cambiare, spesso ripetendoli, i comandi per gli attributi temporanei in determinate stampe, come per esempio PRINT FLASH 1; BRIGHT 1; ... seguiti poco oltre da PRINT FLASH 0; BRIGHT 0; ... (naturalmente potrebbe trattarsi anche di altri tipi di attributi, modificati spesso in modo temporaneo), può risultare utile assegnare detti attributi ad una variabile stringa: così per es. con LET a$ = CHR$1+CHR$3 si potrà sostituire il primo dei due comandi citati, anche ripetutamente nel programma, con PRINT a$; si può utilizzare un'analoga assegnazione per gli altri attributi che ristabiliscono le condizioni normali.
Vi è anche un modo più sottile ed elegante, anche se meno chiaro (e questo a volte può essere desiderato dal programmatore) per ottenere il medesimo scopo. L'assegnazione ad a$ si farà per esempio così:

- impostare LET a$ = ";
- premere i due SHIFT assieme;
- premere Caps SHIFT + tasto 9 (FLASH 1);
- premere i due SHIFT;
- premere tasto 9 (BRIGHT 1);
- impostare le " di chiusura seguite da ENTER.

Una applicazione non descritta altrove riguarda invece il comando SAVE.
Se può interessare che un programma venga salvato in modo che all'atto del ricaricamento con LOAD il suo

nome spicchi con certi attributi (colore, lampeggio, ...), e magari in una posizione determinata dello schermo (ad esempio caricando un programma composto di più parti, il nome completo venga a comporsi al centro dello schermo a mano a mano che si caricano le varie sezioni di nastro, magari colorando diversamente varie parole, e raggiungendo così nomi più lunghi dei 10 caratteri solitamente ammessi), si può utilizzare il comando SAVE con queste varianti:
a) SAVE CHR$ 16 + CHR$ n + "nome" (dove n è il codice di colore INK, 0 ÷ 7)
oppure SAVE "(modo E) (Caps SHIFT + tasto n) nome"
(naturalmente invece del carattere di controllo 16 per INK ci può essere un altro per controllo di un diverso attributo, o più in fila);
b) SAVE CHR$23 + CHR$ c + "nome" (dove c è il numero di colonna a partire da cui verrà stampato il "nome")
oppure SAVE CHR$ 23 + "(c) nome" (numero e nome di seguito, senza spazi);
c) SAVE CHR$ 22 + CHR$ r + CHR$ c + "nome" (dove r, c sono il numero di riga e di colonna della posizione da cui verrà stampato "nome").
Nei casi b) e c) naturalmente si potrà anche aggiungere il/i carattere/i di controllo per gli attributi, come in a).
Provate qualcuno di questi SAVE (badate solo che il nome va accorciato per non superare i 10 caratteri complessivi per singolo SAVE) e osservate cosa succede al LOAD!

TEXAS TI99 4/A

Ring

di Sergio Borsani

Il gioco Ring, in versione TI Extended BASIC viene presentato in questa rubrica in quanto consente di illustrare una tecnica semplice, ma efficace di animazione che si basa sulla ridefinizione continua dei caratteri.
Ormai i lettori più smaliziati conoscono le prestazioni del microprocessore grafico TMS9918A che recentemente è stato adottato da più di dieci case costruttrici di micro nello standard MSX. Altresì sono note le possibilità offerte dal modulo TI Extended BASIC per la creazione ed il controllo degli sprite, cioè degli ormai cosiddetti spiritelli o disegni mobili sullo schermo. Già, perché senza gli sprite non sarebbe possibile riprodurre la complessità, la molteplicità e la velocità dei movimenti che generalmente richiede un gioco.

Delle istruzioni grafiche dell'Extended BASIC si è già parlato diffusamente in altri articoli (vedi Gli spiritelli, *Bit* Maggio 1984); in questa sede non è necessario parlare di esse dettagliatamente. Quello che invece merita maggior risalto e che sembra caratterizzante in questo programma è l'ampia definizione e ridefinizione di caratteri grafici per consentire la rappresentazione dei pugili nelle varie posizioni e rendere il più possibile realistiche le immagini in tutte le fasi dell'incontro.
I caratteri grafici definibili dall'utente non sono molti. Essi possono andare dal numero 32 di codice ASCII al numero 143, in tutto 112. Di questi, i primi 96 sono già definiti ed il loro uso per la grafica comporta la rinuncia o, ad esempio, delle lettere minuscole o addirittura di qualsiasi testo scritto. Tenendo poi conto del fatto che gli sprite definiti su una matrice di 8x8 pixel risultano troppo piccoli e che è da preferirsi un pattern 16x16 con una CALL MAGNIFY(3), si comprende come sussistano evidenti limitazioni.
In questo gioco ogni figura è composta da tre sprite aventi le dimensioni di 16x16 pixel. Il primo sprite forma il busto, il secondo le gambe ed il terzo, di diverso colore, i calzoncini e le scarpette. In tutto quindi sono necessari 12 caratteri per ogni figura in una determinata posizione. Il movimento tuttavia richiede la creazione di più sprite per la stessa figura. Se si pretendesse di definire all'inizio del programma tutti i caratteri che in esso verranno usati, si sarebbe costretti a gravi rinunce ed a ricorrere si e no ad un paio di pose stereotipate che renderebbero tremendamente noiosa l'azione. È questo il motivo per cui si è ricorsi alla tecnica, un po' inconsueta, di ridefinire più volte nel corso del programma gli stessi caratteri per mezzo di subroutine. In questo modo è stato possibile attribuire ad ogni pugile otto posizioni per il busto e cinque per le gambe, consentendo la loro visione di fronte, di fianco, da seduti e a terra o con il braccio alzato in segno di vittoria. In tutto sono stati definiti 203 caratteri, ben oltre il numero di quelli utilizzabili contemporaneamente. La tecnica consiste nell'usare un gruppo di caratteri e, durante le pause dell'azione, ridefinire i caratteri di un altro gruppo. Nel momento in cui si usano questi ultimi si definiscono i caratteri di un altro gruppo ancora e così via. È importante che il gioco preveda delle pause naturali, come in questo caso l'intervallo tra un round e l'altro, perché la definizione dei caratteri comporta un certo tempo e non sarebbe possibile farlo durante lo svolgimento dell'azione.
I risultati sono stati così soddisfacenti che, come tocco finale, non si è voluto rinunciare nemmeno alla ragazza che porta il cartello con il numero del round successivo! Manca invece l'arbitro, ma si assicura ugualmente un comportamento ineccepibile da parte dei pugili che non trattengono e non portano mai colpi sotto la cintura.

Dopo l'apparizione del titolo il programma prosegue premendo un tasto qualsiasi. L'utente può indicare a piacimento i nomi del campione e dello sfidante e deve

specificare se usa i joystick oppure no. Un'errata risposta causerebbe un messaggio d'errore durante l'esecuzione del programma. In mancanza di joystick il controllo avviene, naturalmente, tramite la tastiera. Le due alternative conducono ad uno svolgimento dell'azione sostanzialmente analogo: tuttavia i joystick sono raccomandabili non solo in quanto il loro uso è più naturale ed immediato, ma anche perché essi producono movimenti composti. Infatti il controllo da tastiera avviene con quattro tasti (quelli con le frecce per il pugile di sinistra) più uno per tirare un colpo mentre con i joystick valgono anche le altre quattro posizioni intermedie, cioè in alto a destra, in basso a sinistra, ecc. In pratica il pugile, può avanzare e contemporaneamente assumere la guardia bassa oppure retrocedere tirando un colpo, ecc.

Per i joystick valgono le seguenti regole. Alto: guardia alta; basso: guardia bassa; destra: spostamento a destra; sinistra: spostamento a sinistra; bottone di fuoco: parte un colpo alto o basso a seconda della posizione assunta dal pugile. Per la tastiera è previsto l'uso di determinati tasti e precisamente quelli con le frecce più il tasto Q, come bottone di fuoco, per il concorrente di sinistra; i tasti I, J, K, M più il tasto P per il concorrente di destra. Nella istruzione CALL KEY sono state definite le tastiere 1 e 2 con le quali i tasti assumono determinati valori, diversi dai codici ASCII (vedi manuale a pag. 91) e che si ripetono identicamente sulle due metà della tastiera che

Listato 1. *Il programma Ring.*

```
100 REM   R  I  N  G
110 REM ****************
120 CALL CLEAR
130 DISPLAY AT(12,12):"R I N G"
140 CALL HCHAR(9,11,42,13):: CALL VCHAR(
10,11,42,5):: CALL VCHAR(10,23,42,5):: C
ALL HCHAR(15,11,42,13)
150 CALL KEY(0,K,S):: IF S=0 THEN 150
160 CALL CLEAR
170 DISPLAY AT(1,1):"SCRIVI IL NOME DEL
CAMPIONE."
180 DISPLAY AT(2,1):"CHE NE DICI DI BIG
HAMMER?"
190 ACCEPT AT(4,1)SIZE(12):CPM$
200 DISPLAY AT(6,1):"ED ORA SCRIVI IL NO
ME DELLO"
210 DISPLAY AT(7,1):"SFIDANTE." :: ACCEP
T AT(9,1):SFD$
220 DISPLAY AT(12,1):"HAI I JOYSTICK? (Y
/N)" :: ACCEPT AT(12,24)SIZE(1)VALIDATE(
"YN"):R$
230 IF R$="Y" THEN JOY=1 :: GOSUB 2120 E
LSE JOY=0
240 DISPLAY AT(20,1):"BENE. ORA PREMI UN
 TASTO PER" :: DISPLAY AT(21,1):"COMINCI
ARE."
250 CALL KEY(0,K,S):: IF S=0 THEN 250 EL
SE CALL CLEAR
260 CALL CHAR(40,"0102040810204080"):: C
ALL CHAR(41,"8040201008040201"):: CALL C
HAR(42,"FF")
270 CALL CHAR(35,"FFFFFFFFFFFFFFFF")
280 CALL CHAR(43,"0103070F1F3F7FFF"):: C
ALL CHAR(44,"80C0E0F0F8FCFEFF")
290 CALL CHAR(36,"3F7F7D5555554444"):: C
ALL CHAR(45,"FFFFFFFFFFFFFFFF")
300 CALL CHAR(37,"FFFEFCF8F0E0C080"):: C
ALL CHAR(38,"FF7F3F1F0F070301")
310 CALL COLOR(1,7,1):: CALL COLOR(2,12,
1)
320 FOR J=1 TO 6 :: CALL HCHAR(14+J,9-J,
43):: CALL HCHAR(14+J,10-J,45,14+2*J)::
 CALL HCHAR(14+J,24+J,44):: NEXT J
330 FOR J=1 TO 3 :: CALL HCHAR(20+J,3,35
,28):: NEXT J
340 CALL VCHAR(17,3,45,4):: CALL VCHAR(1
2,9,45,4):: CALL VCHAR(12,24,45,4):: CAL
L VCHAR(17,30,45,4)
350 FOR J=1 TO 3 :: CALL HCHAR(11+J,10,4
2,14):: NEXT J
360 FOR J=1 TO 3 :: FOR K=1 TO 5 :: CALL
 HCHAR(16+J-K,3+K,40):: CALL HCHAR(10+J+
K,24+K,41):: NEXT K :: NEXT J
370 GOSUB 1440 :: CALL SPRITE(#10,96,2,1
12,80,#11,100,2,128,80,#9,104,8,128,80)
380 CALL MAGNIFY(3)
390 CALL SPRITE(#20,96,2,112,160,#21,100
,2,128,160,#19,104,9,128,160)
400 DISPLAY AT(1,1):"     CAMPIONATO DEL
MONDO     "
410 DISPLAY AT(3,1):"CAMPIONE:" :: DISPL
AY AT(4,1):CPM$
420 DISPLAY AT(3,15):"SFIDANTE:" :: DISP
LAY AT(4,15):SFD$
430 DISPLAY AT(5,1):"ALT.: 185 CM  ALT.:
 182 CM"
440 DISPLAY AT(6,1):"PESO: 105 KG  PESO:
 102 KG"
450 DISPLAY AT(7,1):"VINTE: 65       VINTE
: 53"
460 DISPLAY AT(8,1):"PERSE: 2        PERSE
: 4"
470 DISPLAY AT(9,1):"PARI:  1        PARI:
 3"
480 FOR TIME=1 TO 2000 :: NEXT TIME
490 CALL HCHAR(1,1,32,288):: CALL COLOR(
1,5,1):: CALL SCREEN(11):: CALL COLOR(2,
12,1)
500 FOR J=1 TO 12 :: CALL HCHAR(J,1,35,1
2-J):: CALL HCHAR(J,13-J,37)
510 CALL HCHAR(J,19+J,38):: CALL HCHAR(J
,20+J,35,12-J):: NEXT J
520 C(1)=80 :: C(2)=160 :: GOSUB 1480
530 CALL SPRITE(#9,128,8,128,C(1)-4,#10,
120,2,112,C(1),#11,124,2,128,C(1)-4)
540 CALL SPRITE(#19,116,9,128,C(2)+4,#20
```

Seguito listato Ring.

```
,108,2,112,C(2),#21,112,2,128,C(2)+4)
550 GOSUB 1550 :: CALL SOUND(1000,-1,1)
560 TT=0 :: P(1),P(2)=0 :: F(1),F(2)=1 :
: ROUND=1 :: PG=2 :: W=-1
570 PG=PG+W
580 IF JOY=1 THEN 660
590 CALL KEY(PG,K,S):: IF S=0 THEN CALL
PATTERN(#PG*10,132-PG*12):: F(PG)=1 :: G
OTO 900
600 IF K=5 THEN X=0 :: Y=4 :: GOTO 670
610 IF K=0 THEN X=0 :: Y=-4 :: GOTO 670
620 IF K=2 THEN X=-4 :: Y=0 :: GOTO 670
630 IF K=3 THEN X=4 :: Y=0 :: GOTO 670
640 IF K=18 OR K=11 THEN 810
650 GOTO 900
660 CALL JOYST(PG,X,Y)
670 IF X=0 AND Y=0 THEN 800
680 IF X=0 THEN 740
690 CALL DISTANCE(#10,#20,DD):: IF DD<60
 AND SGN(X)<>W THEN 740
700 IF C(PG)+X>200 THEN 740
710 IF C(PG)+X>50 THEN C(PG)=C(PG)+X ::
CALL LOCATE(#PG*10,112,C(PG),#PG*10+1,12
8,C(PG)+4*W,#PG*10-1,128,C(PG)+4*W)
720 IF PG=1 THEN Z1=76 :: Z2=80 ELSE Z1=
64 :: Z2=68
730 CALL PATTERN(#PG*10+1,Z1,#PG*10-1,Z2
)
740 IF Y=0 THEN 800
750 IF Y=4 THEN 780
760 F(PG)=2 :: IF PG=1 THEN Z=84 ELSE Z=
136
770 CALL PATTERN(#PG*10,Z):: GOTO 800
780 F(PG)=1 :: IF PG=1 THEN Z=120 ELSE Z
=108
790 CALL PATTERN(#PG*10,Z)
800 CALL KEY(PG,K,S):: IF S=0 THEN 900
810 IF PG=1 THEN 840
820 IF F(PG)=1 THEN Z=132 ELSE Z=140
830 GOTO 850
840 IF F(PG)=1 THEN Z=72 ELSE Z=88
850 CALL PATTERN(#PG*10,Z):: KK(PG)=F(PG
)
860 CALL DISTANCE(#10,#20,DD):: IF DD>60
 THEN 900
870 CALL SOUND(100,-7,10)
880 IF KK(PG)<>F(PG-W) THEN P(PG)=P(PG)+1
890 IF P(PG)>100 THEN 990
900 TT=TT+1 :: IF TT>200 THEN CALL SOUND
(1000,-1,1):: GOTO 1230
910 NN=INT(RND*2)+1
920 IF PG=1 THEN 950
930 IF NN=1 THEN Z1=112 :: Z2=116 :: GOT
O 970
940 Z1=64 :: Z2=68 :: GOTO 970
950 IF NN=1 THEN Z1=124 :: Z2=128 :: GOT
O 970
960 Z1=76 :: Z2=80
970 CALL PATTERN(#PG*10+1,Z1,#PG*10-1,Z2
)
980 W=-W :: GOTO 570
990 IF PG=1 THEN 1040
1000 CALL PATTERN(#9,128,#10,120,#11,124
):: GOSUB 1700
1010 CALL LOCATE(#9,128,C(1)-16,#10,128,
C(1),#11,128,C(1)-16)
1020 CALL PATTERN(#10,96,#11,100,#9,104)
:: CALL LOCATE(#20,96,160,#21,112,164,#1
9,112,164)
1030 CALL PATTERN(#20,108):: GOTO 1070
1040 CALL PATTERN(#19,116,#20,108,#21,11
2):: GOSUB 1660
1050 CALL LOCATE(#19,128,C(2)+16,#20,128
,C(2),#21,128,C(2)+16):: CALL LOCATE(#10
,96,80,#11,112,76,#9,112,76)
1060 CALL PATTERN(#20,96,#21,100,#19,104
):: CALL PATTERN(#10,120)
1070 KD=KD+1 :: P(1)=P(1)-10 :: P(2)=P(2
)-10
1080 FOR TIME=1 TO 1000 :: NEXT TIME
1090 C(1)=80 :: C(2)=160 :: CALL DELSPRI
TE(ALL)
1100 CALL SPRITE(#9,128,8,128,C(1)-4,#10
,120,2,112,C(1),#11,124,2,128,C(1)-4)
1110 CALL SPRITE(#19,116,9,128,C(2)+4,#2
0,108,2,112,C(2),#21,112,2,128,C(2)+4)
1120 IF FLAG=1 THEN 1140
1130 IF KD<5 THEN 900
1140 GOSUB 1440
1150 CALL PATTERN(#10,96,#11,100,#9,104)
:: CALL LOCATE(#11,128,80,#9,128,80)
1160 CALL PATTERN(#20,96,#21,100,#19,104
):: CALL LOCATE(#21,128,160,#19,128,160)
1170 GOSUB 2080 :: CALL PATTERN(#PG*10,6
4,#PG*10+1,68)
1180 CALL HCHAR(14,PG*10+2,62)
1190 IF PARI<>1 THEN 1210
1200 CALL PATTERN(#20,64,#21,68):: CALL
HCHAR(14,22,62)
1210 FOR TIME=1 TO 3000 :: NEXT TIME
1220 CALL DELSPRITE(ALL):: CALL CLEAR ::
 END
1230 CALL PATTERN(#9,128,#10,120,#11,124
):: CALL PATTERN(#19,116,#20,108,#21,112
)
1240 ROUND=ROUND+1 :: IF ROUND>12 THEN 1
410
1250 GOSUB 1740 :: CALL HCHAR(19,5,36)::
 CALL HCHAR(19,28,36)
1260 CALL LOCATE(#10,124,34,#11,140,34,#
9,140,34):: CALL LOCATE(#20,124,208,#21,
140,208,#19,140,208)
1270 CALL PATTERN(#10,132,#11,136,#9,140
):: CALL PATTERN(#20,84,#21,88,#19,92)
1280 FOR TIME=1 TO 2000 :: NEXT TIME
1290 GOSUB 1810 :: ON ROUND-1 GOSUB 1860
,1880,1900,1920,1940,1960,1980,2000,2020
,2040,2060
1300 CALL SPRITE(#25,64,2,90,180,#26,68,
2,106,180,#24,76,14,98,180,#27,80,7,74,1
80)
1310 G=2 :: CALL MOTION(#25,0,-3,#26,0,-
3,#24,0,-3,#27,0,-3)
1320 FOR TIME=1 TO 50 :: NEXT TIME :: CH
```

così resta divisa in due parti distinte ma con analoga funzione.
Dopo le presentazioni, all'inizio del combattimento, i concorrenti non devono far altro che avvicinarsi e cominciare a darsele di santa ragione premendo i bottoni di fuoco; i punti vengono assegnati in modo imparziale dal programma con un criterio fin troppo semplice: ottiene un punto un colpo alto che trova l'avversario in posizione bassa o viceversa un colpo alla figura che trova l'avversario in guardia alta. A proposito, non dimenticate di alzare il volume del televisore perché non mancano gli effetti sonori e attenti a non maltrattare troppo le levette dei joystick che non paiono particolarmente robuste!

Descrizione del listato

100-150 - Titolo.
160-250 - Richiesta dei nomi dei pugili e dell'uso dei joystick.
260-310 - Definizione dei caratteri e dei colori per la scenografia.
320-480 - Stampa dei preliminari con la presentazione dei pugili.
490-560 - Inizio del combattimento.
570-980 - Ciclo per il controllo del movimento.
590-650 - Comandi per il movimento tramite la tastiera.
660 - Istruzione per interpretare i segnali del joystick.

La J.*soft*
CERCA
per la nuova rivista Super Apple
un redattore da assumere,
con ampia e provata
conoscenza dei sistemi Apple.
Si richiede la buona
conoscenza della lingua inglese,
residenza Milano/dintorni.
Inviare curriculum completo a
J.*soft* - Via Rosellini, 12
20124 MILANO

Seguito listato Ring.

```
=70+G :: CALL PATTERN(#26,CH)
1330 CALL POSITION(#25,SY,SX):: IF SX>70
 THEN G=-G :: GOTO 1320
1340 CALL MOTION(#25,0,0,#26,0,0,#24,0,0
,#27,0,0)
1350 FOR TIME=1 TO 1000 :: NEXT TIME ::
CALL DELSPRITE(#27,#24,#25,#26)
1360 GOSUB 1480 :: CALL PATTERN(#10,120,
#11,124,#9,128):: CALL PATTERN(#20,108,#
21,112,#19,116)
1370 CALL LOCATE(#10,112,80,#11,128,76,#
9,128,76):: CALL LOCATE(#20,112,160,#21,
128,164,#19,128,164)
1380 CALL HCHAR(19,5,45):: CALL HCHAR(19
,28,45):: CALL SOUND(1000,-1,1)
1390 CALL SOUND(1000,-1,1)
1400 GOSUB 1550 :: TT=0 :: W=-W :: C(1)=
80 :: C(2)=160 :: GOTO 570
1410 IF P(1)=P(2)THEN PG=1 :: PARI=1 ::
FLAG=1 :: GOTO 1090
1420 FLAG=1 :: IF P(1)>P(2)THEN PG=1 ::
GOTO 1090
1430 PG=2 :: GOTO 1090
1440 CALL CHAR(96,"01030303013D7F7F7F7F6
F6F6747477780C0C0C080BCFEFEFEFEF6F6E6E2E
2EE")
1450 CALL CHAR(100,"F7D7C777060606060606
060606060E3EEFEBE3EE6060606060606060606
707C")
1460 CALL CHAR(104,"0707070707070000000
000606060E3EE0E0E0E0E0E0000000000606060
707C")
1470 RETURN
1480 CALL CHAR(108,"0000000030707070 3"&
RPT$("0",16)&"183C3C3C9CBE9FFFFF7F1F1E1E
1E1E")
1490 CALL CHAR(112,RPT$("01",16)&"F0F0F0
F0F0F0F0F0B098BC8C88888888")
1500 CALL CHAR(116,"0101010101010000000
0000010103FB0B0B0B0B0B0000000000000 8C8C
8E98")
1510 CALL CHAR(120,"00183C3C3C3971F9FFFF
FEF8787878780000000C0E0E0E0C0"&RPT$("0"
,14))
1520 CALL CHAR(124,"0F0F0F0F0F0F0F0D19
313111111111"&RPT$("80",16))
1530 CALL CHAR(128,"0D0D0D0D0D0D00000000
0000313171 9C0C0C0C0C0C0000000000000B0B0
C0F0")
1540 RETURN
1550 CALL CHAR(132,"0000000060F0FF7F"&RP
T$("0",16)&"0C1E1E1E0C3EFFFF3F3F3F1F1E1E
1E1E")
1560 CALL CHAR(136,"0000000030707071B3C
3C1C060301000000000000B0B0F0FCFEFEFFFFDF
8F0F")
1570 CALL CHAR(140,RPT$("0",20)&"317F7F3
0000000307878783C1E3F7FFFFFBF1E1E1E1E")
1580 CALL CHAR(64,"0101030307070F0E1C181
```

Seguito listato Ring.

```
81B1B1B1B1B1B0F0F0F0F8F8F838381B0C0C0C0C0C0
C0C")
1590 CALL CHAR(68,"0101030306060000000000
000181838F870707078F8F800000000000000C0C0
E0E")
1600 CALL CHAR(72,"30787878307CFFFFFCFCF
CF878787878000000000060FFFFE"&RPT$("0",16
))
1610 CALL CHAR(76,"0F0F0F1F1F1F1C1C183030
3030303030808C0C0E0E0F0703B1B1B1B1B1B1B1
818")
1620 CALL CHAR(80,"0E0E0E1E1F1F000000000
0003030707080B0C0C060600000000000001818 1
C1F")
1630 CALL CHAR(84,"000000000001010F3F7F7
FFFFFFBF1F000000000C0E0E0E0D83C3C3860C0B
000")
1640 CALL CHAR(88,"000C1E1E1E3C70FCFEFFF
FFD78787878"&RPT$("0",20)&"BCFEFE0C0000"
)
1650 RETURN
1660 CALL CHAR(96,"0000000000000000307070
70F1E1E1F0F000000001F3F7FFFFFFCB0B03030F
0F0")
1670 CALL CHAR(100,"000000000B0F0F8FCFCFC
FCFCFCE3FFFC"&RPT$("0",32))
1680 CALL CHAR(104,"000000000000083CDCDC
DCDCDCCO"&RPT$("0",28)&"60F0F80B")
1690 RETURN
1700 CALL CHAR(96,"00000000F8FCFEFFFF3F0
D0D0C0C0F0F000000000000C0E0E0E0F07B78F
8F0")
1710 CALL CHAR(100,RPT$("0",24)&"060F1F1
0000000000010F1F3F3F3F3F3FC7FF3F")
1720 CALL CHAR(104,RPT$("0",24)&"060F1F1
000000000000103C3B3B3B3B3B030000")
1730 RETURN
1740 CALL CHAR(84,RPT$("0",26)&"030F1E00
000000307878783037C7EFEFE3E3E")
1750 CALL CHAR(88,"1C0F0F0F0C0C0C0C000000
0000000000003EFEFEFE"&RPT$("0",24))
1760 CALL CHAR(92,"0000000000000C0C1C7C0
000000000000003A760E7E7C"&RPT$("0",22))
1770 CALL CHAR(132,"000000000C1E1E1E0C1C
3E7E7F7F7C7C"&RPT$("0",26)&"C0F078")
1780 CALL CHAR(136,"7C7F7F7F"&RPT$("0",2
4)&"38F0F0F030303030383E0000000000")
1790 CALL CHAR(140,"5C6E707E3E"&RPT$("0"
,34)&"303038 3E000000000000")
1800 RETURN
1810 CALL CHAR(64,"10180804040203010103O
70703010101020204E4E8F870F8D890B080B080C
0E0")
1820 CALL CHAR(68,"0101030307060C0C0C080
8080808C1424E0E0C0C0C0C0C0C0603030202020305
090")
1830 CALL CHAR(72,"0101010303030707070303
0101010304E0E0E0C0C0808080808080C0C0302BA
0A0")
1840 CALL CHAR(76,"000008080C06030303"&RPT
$("0",26)&"80C0E060200000000000")
1850 RETURN
1860 CALL CHAR(80,"FFFFF8F0E3E3FFFFFCF1E
3E3E0E0FFFFFFFF1F0FC7C7C70F1F7FFFFF0707F
FFF")
1870 RETURN
1880 CALL CHAR(80,"FFFFF8F0E3E3FFFFFFFFE
3E3F0F8FFFFFFFF1F0FC7C7C70F0FC7C7C70F1FF
FFF")
1890 RETURN
1900 CALL CHAR(80,"FFFFFFFFFEFCF9F3E7E0E
0FFFFFFFFFFFFFF8F0F0F8F8F8F8F07078F8F07F
FFF")
1910 RETURN
1920 CALL CHAR(80,"FFFFE0E0E7E7E0E0FFFFE
7E7F0F8FFFFFFFF0707FFFF1F0FC7C7C7C70F1FF
FFF")
1930 RETURN
1940 CALL CHAR(80,"FFFFF8F0E3E3E0E0E3E3E
3E3F0F8FFFFFFFF1F0FC7FF1F0FC7C7C7C70F1FF
FFF")
1950 RETURN
1960 CALL CHAR(80,"FFFFE0E0E7FFFFFEFCF8F
8F1F1F1FFFFFFFF0707C78F1F3F7FFFFFFFFFFF
FFF")
1970 RETURN
1980 CALL CHAR(80,"FFFFF8F0E3E3E3F0F0E3E
3E3F0F8FFFFFFFF1F0FC7C7C70F0FC7C7C70F1FF
FFF")
1990 RETURN
2000 CALL CHAR(80,"FFFFF8F0E3E3E3E3F0F8F
FE3F0F8FFFFFFFF1F0FC7C7C7C70707C7C70F1FF
FFF")
2010 RETURN
2020 CALL CHAR(80,"FFFFF3E3C3F3F3F3F3F3F
3F3F3E1FFFFFFFF8703333333333333333330387F
FFF")
2030 RETURN
```

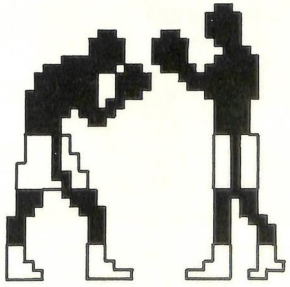


Figura 1. *Per ottenere un'azione più ricca e realistica sono stati definiti più volte, nel corso del programma, gli stessi caratteri per un numero complessivo di 203 caratteri, ben superiore quindi a quelli normalmente disponibili.*

670-790 - Controllo della posizione dei pugili sul ring, della loro distanza reciproca e della posa assunta.
800 - Controllo dei "bottoni di fuoco" dei joystick o dei tasti programmati per far partire i colpi ai pugili.
810-870 - Controllo della forma assunta dai pugili nell'atto di colpire l'avversario. Se il colpo è andato a segno, la linea 870 produce un rumore.
880-890 - Punteggio assegnato dal programma ai pugili in base ai colpi andati a segno. Se un contendente subisce più di 80 colpi subisce un atterramento.
900 - Controllo del tempo per determinare la durata dei round.

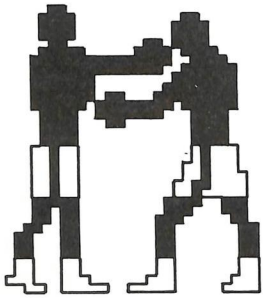


Figura 2. *Il controllo dei movimenti avviene o tramite tastiera o meglio con i joystick. Manca l'arbitro, tuttavia i pugili si comportano in modo molto corretto, non trattengono né colpiscono sotto la cintura.*

Seguito listato Ring.

```
2040 CALL CHAR(80,"FFFFF3E3C3F3F3F3F3F3F
3F3F3E1FFFFFFFFCF8F0FCFCFCFCFCFCFCFCF87F
FFF")
2050 RETURN
2060 CALL CHAR(80,"FFFFF3E3C3F3F3F3F3F3F
3F3F3E1FFFFFFFF870333F3F3E7CF9F3F3F0303F
FFF")
2070 RETURN
2080 CALL CHAR(64,"0103030313D7F7F7F7F6
F6F67474747BCCCCCDC9CBCF8F8F0F0F0F0E0E0E
0E0")
2090 CALL CHAR(68,"F7D7C7770606060606060
60606060E3EE0E0E0E0606060606060606060607
07C")
2100 CALL CHAR(62,"1C1C1C0808080808")
2110 RETURN
2120 DISPLAY AT(14,1):"JOYSTICK 1 PUGILE
 DI DESTRA"
2130 DISPLAY AT(16,1):"JOYSTICK 2 PUGILE
 A SINISTRA"
2140 DISPLAY AT(18,1):"RILASCIA IL TASTO
 ALPHA LOCK"
2150 RETURN
```

910-970 - Istruzioni per produrre un certo movimento nei pugili anche in assenza di istruzioni specifiche.
980 - Assegnazione del controllo alternativamente ai due pugili e rinvio all'inizio del ciclo.
990-1130 - Routine per l'atterramento di un pugile. La linea 1070 modifica il punteggio per consentire la ripresa dell'incontro ma dopo 5 atterramenti la linea 1130 fa terminare l'incontro.
1140-1220 - Routine per il termine dell'incontro. Si alza il braccio del vincitore.
1230-1400 - Routine per l'intervallo tra un round e l'altro. I pugili si siedono e compare la ragazza con il numero del round successivo.
1410-1430 - Controllo del punteggio al termine dei 12 round.
1440-1470 - Subroutine per la definizione dei caratteri per rappresentare i pugili di fronte.
1480-1540 - Definizione dei caratteri per rappresentare i pugili nella posizione base.
1550-1650 - Definizione dei caratteri per rappresentare i pugili nelle varie pose assunte durante il combattimento.
1660-1690 - Definizione dei caratteri per il pugile di destra al tappeto.
1700-1730 - Definizione dei caratteri per il pugile di sinistra al tappeto.
1740-1800 - Subroutine per definire i caratteri che rappresentano i pugili seduti.
1810-1850 - Subroutine per la definizione dei caratteri che rappresentano la ragazza che porta il cartello con il numero del round.
1860-2070 - Varie subroutine, richiamate dalla linea 1280, per definire ad ogni round i caratteri per il cartello con il numero corrispondente.
2080-2110 - Definizione dei caratteri per rappresentare i pugili con il braccio alzato in segno di vittoria.
2120-2150 - Messaggio sull'uso dei joystick.

Principali variabili

CPM$, SFD$ - Nome del campione e dello sfidante.
JOY - Variabile flag per indicare l'uso dei joystick.
C(1), C(2) - Contengono i numeri di colonna, in pixel, della posizione degli sprite rappresentanti i pugili.
TT - Variabile per il controllo del tempo. Viene incrementata di un'unità ad ogni ciclo.
P(1), P(2) - Punteggi dei due contendenti.
F(1), F(2) - Atteggiamento o posizione assunta dai pugili. Il valore 1 indica una guardia alta mentre il valore 2 una guardia bassa.
ROUND - Memorizza il numero del round.
PG - Assume i due valori, 1 o 2, in riferimento al pugile cui spetta il controllo in un dato istante.
W - Variabile di controllo. Ad ogni ciclo cambia il suo valore da -1 a +1 controllando il contenuto della variabile PG.

FACILI CAMBIAMENTI

COLORE DEGLI ELEMENTI SCENOGRAFICI. Con un apparecchio televisivo in bianco e nero si impone l'uso di colori contrastanti dal momento che molti risultano quasi indistinguibili tra di loro. Con un televisore a colori invece si possono utilizzare tutti gli abbinamenti con i 16 colori. In questo caso si devono modificare le linee 310 e 490.

COLORE DEI PUGILI. Essendo questi degli sprite, a nulla servono le istruzioni CALL COLOR. Il colore, in questo caso, è definito dalla terza costante contenuta tra parentesi nell'istruzione CALL SPRITE. Si ricorda che il pugile di sinistra è associato agli sprite #10 e #11, mentre i suoi calzoncini sono lo sprite #9; il pugile di destra è dato invece dagli sprite #20 e #21, mentre i suoi calzoncini sono lo sprite #19. Pertanto le linee da modificare sono tutte quelle che ad essi si riferiscono.

DURATA DEI ROUND. La durata dei round è determinata dall'istruzione IF THEN contenuta alla linea 900. Attualmente ogni ripresa termina quando TT raggiunge il valore 200. Scrivendo al suo posto un numero maggiore, ovviamente, si allunga la durata e viceversa.

DETERMINAZIONE DEL PUNTEGGIO. Attualmente il punteggio è stabilito alla linea 880. Semplicemente si assegna un punto se viene tirato un colpo alla figura mentre l'avversario è in guardia alta, oppure se un colpo al volto trova l'avversario in guardia bassa. Si può sostituire la linea 880 con una routine che determini il punteggio in base a criteri più complessi.

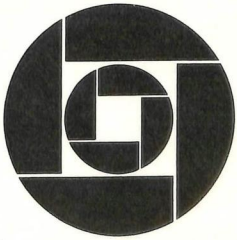
PRIMO ATTERRAMENTO. Il primo atterramento si verifica quando uno dei due pugili abbia subito 100 colpi validi ai fini del punteggio. Generalmente ciò non si verifica prima di 8-10 riprese. Se si desidera accelerare i tempi basta modificare la linea 890 scrivendo, ad esempio: IF P(PG) >50 THEN 990.

NUMERO DEGLI ATTERRAMENTI. Quando si verifica un atterramento i punteggi dei due pugili vengono diminuiti entrambi di 10 unità e l'incontro prosegue. Dopo 5 atterramenti, di uno o di entrambi i pugili, l'incontro termina per KO. Cambiando la linea 1130 si può consentire un numero maggiore o minore di atterramenti.

X, Y - Variabili di stato dei joystick. In base al loro valore si produce una particolare azione.
Z1, Z2 - Variabili di lavoro contenenti i codici ASCII dei caratteri definiti via software, rappresentanti i due pugili.
TIME - Variabile di controllo dei loop per creare delle pause.
KD - Numero degli atterramenti subiti dai due pugili.
FLAG - Variabile flag che assume il valore 1 al termine delle 12 riprese. A differenza del KO, in questo caso, è previsto il pareggio.
K, S - Usate nell'istruzione CALL KEY, indicano rispettivamente il numero di codice ASCII del tasto premuto e lo stato della tastiera.
J - Variabile di lavoro per il controllo dei cicli FOR NEXT.

Figura 3. *La tecnica usata con i caratteri grafici definiti via software consente una certa libertà e limiti meno angusti. Nel programma non si è voluto rinunciare nemmeno alla ragazza che, durante l'intervallo, porta il cartello con il numero del round seguente.*

in un colpo d'occhio...



...il panorama completo di tutte le attività italiane.

l'Annuario Generale Italiano
viene letto da milioni di esponenti politici, governativi, amministrativi, industriali, bancari e finanziari.
Viene consultato da coloro che studiano, riflettono, valutano ed alla fine... decidono.

GUIDA MONACI
ANNUARIO GENERALE ITALIANO

00187 ROMA - Via F. Crispi, 10 - Tel. 06/483.401 - Telex 613462 MONACI
20145 MILANO - Via V. Monti, 86 - Tel. 02/345.8567 - Telex 332849 MONACI

PICCOLI ANNUNCI

Apple

Vendo per Apple II - le **programmi di ogni genere** ad ottimo prezzo. Dispongo di gestionali, giochi, utilità, tra cui: Apple Writer, Personal Data Base, VisiCalc, LOGO, ProDOS, LISP, ingegneria, grafica... Agostino Carollo - Via Lungo Leno Destro, 21 - 38068 Rovereto (TN) - Tel. 0464/31190

Medici utenti Apple contatterebbero colleghi per lo scambio di esperienze col personal computer e per la creazione di un pool di programmi applicativi adatto alle esigenze mediche. Dr. Matteo Pinto - Via Bandiera, 48 - 20099 Sesto S. Giovanni (MI) - Tel. 02/2424537

Compro e cambio **programmi** per Apple II euro. Qualsiasi tipo di programma mi va bene, comprerei anche stampante per Apple in buono stato e modico prezzo. Temistocle Sidoti - Via Candiolo, 41 - 10127 Torino - Tel. 011/6053805

Vendo **programmi per Apple II e Macintosh** altamente professionali, su richiesta anche personalizzazioni. Luigi Palumbo - Via A. Ristori, 8 - 00197 Roma - Tel. 06/802783

Vendo **Apple II plus 48 Kbyte** più unità disco, stampante grafica, varie schede di interfaccia. Dispongo di vario software su diskette. Molti programmi originali. Franco Vagli - Napoli - Tel. 081/7233050

Dispongo di **oltre 400 programmi** per Apple II e compatibili. Desidero effettuare scambi con altri appassionati. Inviatemi la vostra lista e spedirò la mia. Rispondo a tutti. Isabella Bottini - Via Galilei, 681 - 18030 Sanremo (IM) - Tel. 0184/882095

Vendo **Apple II 64 Kbyte**, due drives, monitor 12", stampante Honywell 132 colonne, interfaccia parallela, paddles, PROM-Minus, modulatore TV, vastissimo software professionale con manuali. Tutto a L. 3.100.000. Antonio Balzarini - Via Del Vigna, 189 - 57100 Livorno - Tel. 0586/422276

Causa cambio sistema vendo per Apple II **10 dischetti doppia faccia** con oltre 50 programmi di giochi e utilità (Flight Simulator, Visidex, PFS, ecc.) a L. 100.000. Bruno Rota - Via Passo di Brizio, 6 - 20148 Milano - Tel. 02/408243

Commodore

Per CBM 64 cambio **programmi e giochi solo su cassetta**. Possiedo anche molti listati. Cerco inoltre utenti CBM 64 nella zona di Palermo. Inviatemi le vostre liste o telefonate ore pasti. Umberto Pirozzuolo - Via Gustavo Roccella, 233 - 90128 Palermo - Tel. 091/423314

Cerco **floppy disk drive** per C64 in buone condizioni. Telefonare ore pasti. Roberto Ambrosi - Via Vittorio Alfieri, 19 - 54100 Massa - Tel. 0585/40023

■ Per CBM 64 vendo oltre **1.000 programmi su disco e** ■ ■ **cassetta.** Cambio solamente con nuovi arrivi. Come occa- ■ ■ sione vendo nastro con 20 giochi a L. 50.000. Telefonare o ■ ■ richiedere lista a: Fabio Fontana - Via Caio Canuleio, 136 - ■ ■ 00174 Roma - Tel. 06/7482345 ■

Cambio, vendo **programmi** per CBM 64. Ne possiedo circa 300, tutti di ottima qualità a prezzi bassissimi. Richiedere liste gratuite. Rispondo a tutti, massima serietà. Telefonare o scrivere a: Giuseppe Mascali - Via R. Margherita, 573 - 98028 S. Teresa Riva (ME) - Tel. 0942/791692

Cambio, vendo, compro **programmi** per CBM 64. Annuncio sempre valido, richiedere lista. Assunta Lama - Via G. Rossini, 4 - 19100 La Spezia

Vendo **Turbotape** per CBM 64 a sole L. 15.000. Offro inoltre Pole Position e molti altri giochi a prezzi trattabili. Se interessati spedire la vostra lista giochi. Risponderò con la mia. Luca Boscolo - Via Marinella, 83 - 30028 S. Michele al Tagliamento (VE) - Tel. 0431/57215

Vendo **30 giochi** a L. 5.000 l'uno per CBM 64. Gestionali a prezzi vari. Cedo inoltre libro originale inglese con 25 giochi a L. 35.000. Offerta sempre valida. Cerco contatti con Commodore Club Christian Fassetta - Via Stresa, 48 - 00135 Roma - Tel. 06/3492628

Vendo, **PET/CBM 3032** L. 750.000 o più registratore L. 800.000. Omaggio circa 40 programmi. Tutto in ottimo stato e con manuale in italiano. Telefonare dopo le 20:30. Athos Passarini - Via Zena, 25 - 40065 Pianoro (BO) - Tel. 051/771126

Vendo per VIC 20 **giochi in LM e in BASIC** anche espansi (8 Kbyte). Alcuni giochi:Traxx, Uomo Jet, Pac Man, Frog e molti altri. Invio la lista dei giochi gratuitamente. Roberto Zannoni - Via Biancardi, 9 - 20075 Lodi (MI) - Tel. 0371/53847

Vendo per C64 i **più recenti giochi ed utilities con istruzioni** a prezzi modici. Scrivere o telefonare. Mario Berardo - Via Vittorio Emauele, 165 - 12042 Brà (CN) - Tel. 0172/421777

Compro, vendo **giochi** per Commodore 64 in zona Torino. Angelo Tosti - Via G. Bove, 1 - 10129 Torino - 011/504994

Vendo, causa passaggio a sistema superiore, **VIC 20 con registratore dedicato**, joystick, 3 manuali, 1 cassetta con 40 giochi stupendi, molte riviste e più di 300 programmi. Ottime condizioni. L. 380.000. Giovanni Azara - Via Peonie, 3 - 20089 Rozzano (MI) - Tel. 8240807

Vendesi per C64 **combinatore telefonico** comprendente interfaccia elettronica e software. Prezzo L. 55.000 tutto compreso. Scrivere o telefonare a: Roberto Allario - Via Migliarone, 3 - 10091 Alpignano (TO) - Tel. 011/9672206

Cambio **programmi** per CBM 64. Possiedo molti games, utility. Programmi per velocizzare la lettura/registrazione dati registratore. Cerco programmi duplicatori disco. L'annuncio è sempre valido. Paolo Farri - Via Matteucci, 53 - 41100 Modena - Tel. 059/354025

Cambio, compro, vendo **più di 100 programmi** a prezzi imbattibili, con Turbo (in omaggio), che aumenta di 10 volte il caricamento. Esempi: Decathlon, Zaxxon, Flip Flop, One On One, Magic Desk, ecc. Luca Gandola - Via Marconi, 412 - 18038 Sanremo (IM) - Tel. 62046

Vendo per Commodore 64 **circa 200 giochi**: Koala joystick, Sam Reciter, Turbotape, Screen Graphics, Magic Desk e tanti altri. Il prezzo massimo dei programmi è di L. 10.000. Scrivere o telefonare a: Rosario Cristaudo - Via Mengarini, 11 - 95122 Catania - Tel. 095/474624

Compro, cambio, vendo **programmi** per CBM 64. Solo zona Catania. Fabio Bellassai - Via Cesare Lombroso, 13 - 95100 Catania (CT) - Tel. 095/415353

Vendo, cambio **favolosi video game** per C64. Offerta sempre valida. Dispongo inoltre di Sintetizzatore Vocale. Desidero contattare possessori di linguaggio macchina. Scrivere o telefonare. Armando Aquino - Via Appia, 8 - 83042 Atripalda (AV) - TEL 0825/626498

Vendo **programmi di giochi, utilità e gestionali** per Commodore 64 (oltre 1.500), Apple, Spectrum e QL. Vendo interfaccia programmabile per Spectrum e modem per Commodore 64. Maurizio Carola - Via L. Lilio, 109 - 00143 Roma - Tel. 06/5917363

Cerco per C64 **Pascal con manuale**. In cambio offro fino a 10 programmi (Decathlon, Sam Reciter, Easy Script, Baseball, Buck Rogers, DataBase, Manager, Dr. J. & Larry Bird Go One On One ed altri 100). Sergio Di Nardo - Via Monte Ve'ino, 15 - 67100 L'Aquila - Tel. 0862/61633

1.000 programmi per Commodore 64 e Spectrum ti aspettano computer Club - Casella Postale 7220 - 16166 Genova Quinto

Attenzione! Vendo e cambio **fantastici programmi** dal prezzo bassissimo, scrivetemi! Risposta assicurata! Flavio Chianese - Via Virgilio, 17 - 34170 Gorizia - Tel. 0481/3318

Vendo **più di 300 programmi su nastro e disco** per C64 a L. 2.000 - 5.000. Posseggo ultimissime novità, disposto anche allo scambio. Per ulteriori informazioni rivolgersi a: Alessandro Grossi - Via Dario Campana, 19 - 47037 Rimini (FO) - Tel. 0541/773458

Vendo **programmi** per C64, giochi e utilità in Turbo Tape. Alcuni esempi: Pac Man, Zaxxon, Magic Desk, Simon BASIC. Richiedere lista, annuncio sempre valido. Luigi Bettini - Via Venezia, 121 - 30039 Strà (VE)

Vendo, cambio **più di mille programmi** per Commodore 64, quasi tutti i videogiochi esistenti più moltissime utility e programmi gestionali. Prezzi imbattibili; scrivere o telefonare dopo le 19. Marco Coletti - Via Principe, 56 - 38010 Mollaro (TN) - Tel. 0463/39181

Compro o cambio **qualsiasi programma su cassetta** o solo listato per C64. Inviare eventuali liste con prezzi. Annuncio sempre valido. Scrivetemi. Marco Costantino - Via Trento, 63 - 38086 Pinzolo (TN) - Tel. 0465/51610

Vendo **oltre 250 programmi accuratamente selezionati** per C64 90% in LM, non copiati da riviste su disco o cassetta. Esempio: Decathlon, One On One). Prezzi da L. 4.000 a L. 13.000. Severino Mondany - Via Versaglio, 8 - 26010 Crema

Cambio e vendo **software 64**, posseggo molti programmi anche di utilità, vendo a prezzi modicissimi. Inviate la vostra lista ed io invierò la mia. Massima serietà, programmi ottimi. Massimiliano Prunesti - via Alzavole, 20 - 00169 Roma - Tel. 06/2678447

Vendo oppure cambio **programmi** per 3000 - 4000 - 8000 aperti con manuale. Ho anche programmi e giochi per 2000 e programmi e giochi per C64, con ultime novità dalla Germania e dagli USA. Giuseppe Dallago - Via Stradone Vecchio, 12 - 38019 Tuenno (TN) - Tel. 40260

Cambio **software di ogni genere** per Commodore 64, esclusivamente di persona, onde evitare truffe, presso il mio domicilio o altrui nella zona dei Castelli Romani. Floppy o cassetta, esclusa vendita. Silvano Funghi - Via Cola di Rienzo, 5 - 00047 Marino Laziale (RM) - Tel. 06/9386320

Il Club 64 di Torino cerca contatti in tutt'Italia. Abbiamo circa 3.000 programmi, spediteci la vostra lista e ricevereta la nostra. Oppure telefonate per informazioni. Virga Giovanni - Via Fratelli Garrone, 67/36 - 10127 Torino - Tel. 011/6061336

Cambio, vendo **software** per Commodore 64. Disponibili oltre 600 programmi giochi, utility e gestionali. Serietà. Cerco stampante MPS801 o Seikosha 550. Claudio Larise - Via Pietro Micca, 10 - 13051 Biella (VC)

Cambio, vendo per C64 **oltre 150 programmi di giochi** e molti di utility. Per l'eventuale invio di liste scrivere o telefonare ore pasti. Si assicura la massima serietà e prezzi bassissimi. Carmine Parrella - Via T. Mommsen, 2 - 82100 Benevento - Tel. 0824/48068

Vendo per Commodore 64 **200 programmi tra i più belli in circolazione:** Koala Joystick, Sam Reciter, Mundial Soccer, Copy Cartucce, Turbotape, Pole Position Atari e tanti altri. Scrivere o telefonare. Rosario Cristaudo - Via G. Mengarini, 11 - 95122 Catania - Tel. 095/474624

Cambio/vendo a prezzi stracciati **programmi** per C64. Dispongo di moltissimi giochi in LM e 3D, utilities e gestionali con manuali in inglese e in italiano. Inviare e richiedere la lista. Roberto Franzoso - Piazza Medaglie d'Oro, 6 - 14100 Asti

Vendo e cambio **ottimi programmi originali** per C64 a prezzi bassissimi. Annuncio sempre valido. Gaetano Pagano - Via Papa Giovannni XXIII, 2 - 80041 Boscoreale (NA) - Tel. 081/8584933

Compro, cambio, vendo **molti programmi** per Commodore 64. Richiedere il listino, prezzi da regalo. Per acquisti superiori alle L. 50.000 ulteriori sconti. Cerco manuale in italiano del programma Chess 7.0. Fabrizio Rizzi - Castello 3060 B - 30122 Venezia - Tel. 041/22883

Vendo **Extended BASIC** CBM 64 completo di istruzioni a L. 40.000. Pietro Nieddu - Via Bianchi, 1 - 07100 Sassari

Cambio **oltre 800 programmi** per Commodore 64. Offro massima serietà. Rispondo a tutti. Inviare liste e proposte a: Fabrizio Raucci - Via Terme di Traiano, 38 - 00053 Civitavecchia (RM) - Tel. 0766/29988

Cambio, vendo **programmi** per VIC 20 inespanso. Inviate o richiedete lista. Possibilità di scambi a mano nella zona di Novara o Milano. Cassetta con 20 programmi a scelta a L. 20.000 spese incluse. Paolo Beltramini - Via D'Enricis, 38 - 28100 Novara (NO) - Tel. 0321/457441

Cambio, vendo **programmi di vario genere** per CBM 64. La maggior parte è velocizzata tramite "Turbotape". Presenti nelle liste programmi di notevole interesse (S.A.M., Simon, Panorama HR, EXBASIC, ecc.). Paolo e Massimo - c/o Selenia Spazio S.p.A. - Tel. 43682111

Serve un programma o una procedura realizzata, su Commodore 64 o Olivetti M20, da un programmatore professionista Allora, solo per la Lombardia: Silvestro Gabba - Via R. Rossi, 54/i - 27020 Dorno (PV) - Tel. 0382/848085

Vendo **VIC 20** più 16 Kbyte RAM, 8 Kbyte RAM, 3 Kbyte RAM, Hi-Res, Vic Mon, Tool Kit, Mother Board, Programmatore di EPROM. Vendo programmi per C64. Inoltre vendo 6116 C-DOS. Telefonare ore 19-20. Nicola Pedroli - Casella Postale - 20080 Cisliano (MI) - Tel. 901887

Attenzione! Appello rivolto a tutti gli utenti di C64 umbri: **è nato a Foligno il "Folignòs 4C"**, il primo Commodore Club umbro. Sede centrale a Foligno, sedi in allestimento in Umbria. Folignòs 4C - Voc. S. Andrea, 17 - 06031 Bevagna (PG) - Tel. 0742/72591

■ Vendo per Commodore 64 **1 videogame più 1 floppy disk** ■ ■ per L. 20.000 e altro software a prezzi convenienti. Telefo- ■ ■ nare ore pasti. Maurizio Coccorese - Via Gentile, 108/D - ■ ■ 70126 Bari - Tel. 080/491374 ■

Cambio e vendo **meravigliosi giochi** per VIC 20. L'annuncio è sempre valido, scrivetemi o telefonatemi, inviandomi le vostre liste. Ciao, vi aspetto. Giovanni Tatangelo - Via Gramsci, 15 - 26013 Crema - Tel. 0373/81146

Vendo, per passaggio ad altro sistema, **Commodore 64 più registratore**, joy (aventi 4 mesi di vita), giochi in LM tra cui: Jump Man, Pit Stop, Donkey Kong, Calcio, ecc., Turbotape, il tutto a L. 650.000 trattabili. Giuseppe Maio - Via S. Gaetano, 5 - 82100 Benevento

Siamo due ragazzi che hanno il VIC 20 e lo Spectrum. Vendiamo a prezzi bassissimi **programmi di qualsiasi tipo**. Risposta assicurata! Cerchiamo disperatamente "Chiave" per VIC 20. Luca Maccheroni - Via Virgilio, 27 - 34170 Gorizia - Tel. 0481/89110

Cambio, vendo **moltissimo software** per CBM 64 tra cui: Hulk Match 2, Pitfall 2, Hero, Space Taxi, Girus, Bruce Lee, Flight Simulator 2 con istruzioni in italiano. Telefonatemi o scrivetemi, rispondo a tutti. Dante Psanello - Via Catalani, 63 - 20131 Milano - Tel. 02/2829512

Per Commodore 64 e Sinclair Spectrum, disponendo di un notevole archivio software vendo **cassetta con 5 giochi a scelta** a L. 10.000. Disponibile anche su disco. Annuncio sempre valido. Ivano Parbuono - Via A. di Cambio, 4 - 37138 Verona - Tel. 045/568649

Vendo **VIC 20** più registratore, joystick, manuale "Alla scoperta del VIC 20", 10 cassette di videogiochi, cavetti, libro istruzioni. Tutto usato pochissimo e in perfetto stato a sole L. 260.000. Giuseppe Zitarosa - Via Cuniolo, 12 - 15057 Tortona (AL) - Tel. 0131/81234

PICCOLI ANNUNCI

Vendo **Mailing List scritto interamente in LM**. E' originale perché e scritto da me. Possiede ottime caratteristiche (è in italiano). Gira su SX 64 e CBM 64 più disco. Andrea Rigoni - Via Giuseppe Verdi, 16 - 35031 Abano Terme (PD) - Tel. 049/811447

Compro, vendo, cambio **programmi** per Commodore 64. Inviare lista, risponderò a tutti. Annuncio sempre valido. Alfredo Benincasa - Via Buccini, 4 - 81100 Caserta (CE) - Tel. 0823/324778

Per cessazione sistema vendo **programmi** per CBM 64. Giochi: Donkey Kong, Pole Position, Triad, ecc utility: mutuo, gestione, magazzino, futuri investimenti, ecc. Invio lista gratuitamente. Guido Nocciolini - Via P. Nenni, 6 - 53100 Siena - Tel. 288564

Scambio **programmi** per CBM 64. Ne possiedo più di 300, tra giochi e utility. Inviatemi la vostra lista, rispondo a tutti con la mssima serietà. Annuncio sempre valido. Claudio Bosso - Via Danimarca (Marano), 4 - 80016 Napoli - Tel. 081/7423570

Eccezionale! Vendo **migliaia di programmi** per Commodore 64. Giochi come: Circus Charlie, Hyper Olimpics, Dracul centinaia di utility come: Totoprofessional, ecc. Telefonatemi! Domenico Gallè - Casella Postale, 36 - 88029 Serra San Bruno (CZ) - Tel. 0963/71210

Cambio, vendo **software** per C64 tra cui Decathlon (Activision), Pole Position (Atari), Enduro (Epyx), ecc. Inviare liste. Inoltre cerco manuale del Simon's BASIC (anche in inglese). Massimo Fuoco - Via Q. Ennio, 70 - 72021 Francavilla Fontana (BR) - Tel. 0831/941736

Vendo **Commodore con registratore** e due joystick, Simon's BASIC con 114 comandi, Turbo Tape, diversi giochi (Calcio, Decathlon, Pitfall 2, ecc), guida al computer a L. 900.000. Lorenzo Longagna - Via Piave, 20-7 - 17100 Savona

Vendo **software** per C64: Buck Rogers, Lazarian, Decathlon, Quix, Seafox, Skramble, Zaxxon, Pole Position, Larry Bird, Forbidden Forest, Dig Dug, Arcadia, Pacman, Simon's BASIC, equazioni, Turbotape a L. 30.000. Andrea Screti - Via Francesco Durante, 7 - 00151 Roma - TEl. 06/5313666

Sinclair

Si è costituito il **Computer Club Ciriè** per utenti Sinclair. Oltre 500 programmi, libri e riviste a disposizione dei soci. Per informazioni scrivere allegando bollo o telefonare. Pasquale Speranza - Loc. Vastalla, 80 - 10073 Ciriè (TO) - Tel. 011/9236137

Possiedi uno Spectrum? Allora leggi bene questo annuncio, inviami la lista dei tuoi programmi e riceverai immediatamente la mia. Creeremo un rapporto di scambio. Fabrizio Martano - via Don Sturzo, 7 - 58100 Grosseto

Cerco possessori di Sinclair ZX81 nella mia città o zone limitrofe per scambio informazioni e listat cerco anche programmi per ZX81 da 16 Kbyte RAM. Scrivere e/o telefonare a: Vincenzo Parla - Via Dant, 63 - 90141 Palermo - Tel. 091/324552

Vendo **Sinclair ZX Spectrum** 48 Kbyte con tastiera professionale, interfaccia Kempston, presa monitor, manuale italiano, 5 libri, fotocopie riviste, manuali vari e 35 nastri con 500 programmi. Contattatemi! Salvatore Ferrante - Viale Regina Margerita, 59 - 98100 Messina - Tel. 090/45033

Amici utilizzatori dello Spectrum **è nato il Bit Sinclair Club** per Torino, Orbassano e tutta la cintura. Diventiamo amici! Sergio Battaglini, 25 - 10126 Torino - Tel. 011/6942208

Vendo per Spectrum **interfaccia joystick programmabile**. Prezzo trattabile, per informazioni o ordini telefonare a: Andrea Tron - Via Raviolo, 8 - 10064 Pinerolo (TO) - Tel. 73442

Vendo **Spectrum 80 Kbyte**, interfaccia programmabile, Superface, tastiera Newnevel, 4 libri in italiano, interfaccia Kempston con joystick, materiale vario, oltre 250 programmi, serie completa di Run - Load'Run più program. Antonio Rescaldani - Via Roma, 50 20024 Garbagnate (MI) - Tel. 02/9958867

Tutti vendono computer, noi vendiamo **i migliori programmi** per Spectrum e Commodore 64. Tutta l'ingegneria, la grafica, gestionali, giochi, copiatori fantastici. Richiedere lista gratuita o telefonare. Giordano Bifolchi - Via G. Nel Corso, 111 - 53045 Montepulciano (SI) - Tel. 0578/757907 716397

Vendo **software** per Spectrum 48 Kbyte, tratto solo con la zona di Catania. Fabio Bellassai - Via Fratelli Bandiera, 13 - 95100 Catania - Tel. 095/415353

Vendo **Sinclair Spectrum 48 Kbyte quasi nuovo** più Interface 1, microdrive, numerosi programmi e riviste a L. 600.000. Telefonare. Franco Guerrini - Str. S. Fermo, 17 - 37121 Verona - Tel. 045/39464

Cedo, causa passaggio a sistema superiore, **oltre 200 programmi** per ZX Spectrum a L. 500 per 16 Kbyte e L. 1000 per 48 Kbyte. Solo cassette da minimo 10 programmi per 48 Kbyte e da 20 programmi per 16 Kbyte. Richiedere lista. Ferdinando Miraglia - Via Avellino, 9 - 81100 Caserta - Tel. 0823/444102

Vendo **una espansione a 48 Kbyte per Spectrum 16 Kbyte** (issue 283) composta da banco 8 RAM 32 Kbyte, 4 integrati logici (74LS157 - 74LS00 - 74LS32) più complete istruzioni per il montaggio. Il tutto a L. 60.000. Paolo Chiartano - Via M. Piccoli, 36 - 10081 Castellamonte (TO) - Tel. 0124/581088

Vendo **Sinclair Spectrum 16 Kbyte** uso minimo perfetto e completo di accessori e manuali, più di 30 giochi, gestionali (VuFile e VuCalc), ecc. su cassetta L. 280.000. Arturo Magno - Via A. Diaz, 6 - 39100 Bolzano - Tel. 0471/36075

Cambio **più di 900 programmi** per lo ZX Spectrum. Se hai delle novità e/o più di 400 programmi, scrivimi, allegando la tua lista. Luigi Ballestin - Via Martiri della Libertà, 376/11 - 18038 Sanremo (IM)

Vendo **Sinclair ZX81**, imballaggio originale, manuale in italiano, "66 programmi con lo ZX81" "Giochiamo con ZX81", cassette giochi (1 Kbyte Arcadian e Games1). Tutto a L. 165.000. Michele Rapin - Via Don Bosco, 19 - 60127 Ancona - Tel. 071/896225

Vendo **programmi** per ZX Spectrum 16/48 Kbyte a L. 1.500 cadauno. Molti programmi in omaggio per ordini superiori alle L. 10.000. Si prega di telefonare ore pasti per accordi. Massimo Nucci - Via Chimera, 22 - 52100 Arezzo - Tel. 0575/351195

Cambio, vendo per Spectrum **oltre 230 programmi di ogni tipo**, giochi, utility, adventure. La lista dei programmi è gratuita. Vendo per joystick interfaccia programmabile a tastiera aL. 35.000. Antonio Scarpa - Via Moro, 63/12 - 30035 Mirano (VE) - Tel. 4355254

Texas

Cerco **modulo Extended BASIC** per Texas TI 99/4A. Disposto offrire al massimo L. 90.000. Telefonare a: Vito Angelillo - Via al Castello, 9 - 10090 Buttigliera Alta (TO) - Tel. 011/932013

Causa militare vendo **TI 99/4A**, alimentatore, cavo registratore, modulatore TV, manuale, cassetta giochi (Bioritmi, Master Mind, ecc.), Scacchi 9 livelli (SSI ottimo stato a L. 350.000 trattabili. Giovanni Çolombo - Via G. Rossini, 4 - 24047 Treviglio (BG) - Tel. 0363/41565

Non gettare il tuo TI 99/4A, se il tuo Field-Engineer non può ripararlo. Compro TI 99/4A danneggiati, rotti, inservibili a prezzi da concordare. Compro anche hardware nelle stesse condizioni. Gian Luigi Calzolaro - Via M. della Benedicta, 3/12 - 16010 Rossiglione (GE) - Tel. 010/925447

Vendo **TI 99/4A** più cavo per registratore, cartuccia gioco, joystick Texas, cassetta con 8 programmi, programmi su carta, tutto poco usato a L. 250.000 trattabilissime. Francesco Diamanti - Via Ippolito Nievo, 62 - 00153 Roma - Tel. 06/5897466

Vendo per TI 99/4A **programmi di vario genere in TI BASIC** su cassetta: elettrotecnica, matematica, ragioneria, memorizzazione buste paga, calendario perpetuo... (anche utilizzando maschere video). Telefonare ore pasti. Marco Pedroni - Via Mazzini, 22 - 26100 Cremona - Tel. 0372/37265

Cerco per TI 99/4A **modulo Extended BASIC funzionante**, completo di manuale d'uso e/o giochi su nastro o cartridge. Tratto con Milano e periferia. Marco Riva - Via Jenner, 10 - 201159 Milano - Tel. 02/6883733

Aiuto! Cerco disperatamente **connettore per interfaccia con registratore a cassette** per TI 99/4A, sono disposto a pagare somme astronomiche! Cerco anche Extended BASIC d'occasione e software vario. Francesco Pipitone - Via Isgrò, 79 - 91025 Marsala (TP) - Tel. 0923/952528

Vendo ad ottimo prezzo **coppia joystick** e cartridges Calcio, Scacchi, Dadi, Tombstone City, Parsec e Munch Man, in blocco o separatamente per TI 99/4A. Telefonare ore pasti. Andrea Perota - Via De Gasperi, 47 - 20025 Legnano (MI) - Tel. 0331/54301

Varie

Eccezionale! **La più lunga lista di software** per i computer Sharp MZ-80A/B/K/700. Quasi 800 titoli originali. Dispongo di gestionali, utilities, giochi, interpreti, DOS, CP/M, ecc. Scrivete! Davide Ardizzola - Via Roma - 18016 S. Bartolomeo Mare (IM) - Tel. 0183/401830

Vendo **"Enciclopedia di Elettronica e Informatica" Jackson**, otto volumi rilegati a L. 200.000, spese postali a mio carico. Angiolo Tavanti - Via Felice Battaglia, 6 - 40135 Bologna - Tel. 051/436843

Cambio strabiliante quantitativo di **software** per i computer Sharp MZ 80A/K/B, MZ 700 e Commodore 64. Scrivete tutti e chiedete la lista con oltre 500 titoli per computer. Davide Ardizzola - Via Roma - 18016 S. Bartolomeo Mare (IM) - Tel. 0183/401830

Vendo **macchina da scrivere elettronica Olivetti ET 351** con unità floppy disk e Word Processor e con quattro margherite. Usata pochissimo, L. 5.300.000. Claudio Impellizzeri - Via G. Ferrari, 11 - 10124 Torino - Tel. 011/87801

Vendo A L. 30.000 (incluse spese postali) **dischetto con programmi d'ogni genere per il micro Z80 N.E.** Spedizione in contrassegno. Telefonare o scrivere. Mario Di Floriano - Via Anello Del Sole, 125 - 33074 Fontanafredda (PN) - Tel. 0434/999071

Vendiamo **software su cassetta** più istruzioni dettagliate e remake dei programmi fra i quali troverete games, utility, didattica. Disponiamo di 5 cassette a L. 7.000 cadauna. Raiti Gabriele - Via Carlo Rosselli, 113 - 96016 Lentini - Tel. 095/945079

Vendo **HP41C più due moduli di memoria** più modulo di matematica a L. 300.00 vendo Texas TI 58C a L. 50.000. Antonio Mormile - Via Tosco Romagnola, 1766 - 56023 Cascina (PI) - Tel. 050/777542

Vendo **consolle Intellevision nuovissima** (Giugno '84) ancora in imballo originale più 3 cartucce: Burger time, Calcio. Space battle. Tutto alla strabiliante somma di L. 400.000 trattabili (valore L. 510.000). Alessandro D'Andrea - Via Principe Umberto, 129/bis - 98100 Messina - Tel. 090/57126

Vendo **pocket computer Casio FX702P** completo di interfaccia FA2, stampante FP10 e numerosi programmi al prezzo di L. 300.000 trattabili. Telefonare ore 19 - 21. Mauro Squaiella - Via O. Vigliani, 201 - 10126 Torino - Tel. 011/616945

Vendo e cambio **programmi di ogni genere**. Posseggo oltre 300 programmi, parecchi appena giunti dagli USA. Accetto iscrizioni per Club negli USA. Roberto Repetti - P.le Barbieri, 3 - 43100 Parma - Tel. 0521/50009

Disponiamo di **ottimi programmi gestionali** per CBM 64 - Hewlett Packard 150. Disponibilità usato, accessori, periferiche CBM 64 e HP150. Richiedere lista a: Pocket Group - Via Amoruso, 34 - 70124 Bari

Vendo **software su cassetta per Casio PB-100** inespansa. Per ricevere l'elenco scrivere a: Fabio Di Donna - Via Cupa Mcedonia, 11 - 80137 Napoli - Tel. 081/290015

Vendo **novità americane** tra cui Bruce Lee, Football manager, Cosmic Kanga, F15 Strike Eagle, Spitfire Ace, Valahalla, e moltissimi altri. Accetto anche scambi. Giuliano Peleggi - Via G. Tornielli, 10 - 00151 Roma - Tel. 06/5280434

Vendo **computer Nuova Elettronica** composta da 40 Kbyte RAM e interfaccia per floppy disk oppure cambio con CBM 64. Ezio Gazzola - Via Gavirate, 16 - 20148 Milano (MI) - Tel. 02/406115

Vendo **programma per Sharp MZ-700**: inc. e spese con 2 cod. po. (bianco e nero) provvisto di spiegazioni registrato su cassetta a L. 15.000. Telefonare ore pasti. Nicola Bellettieri - Via A. Righi, 34 - 50047 Prato (FI) - Tel. 583039

Vendo **amico 2000 48 Kbyte RAM** interfaccia seriale parallela opzionale grafica, tastiera standard, 2 disk driver, monitor b/n 9"; tutto a L. 2.200.000. Vendo inoltre mini robot Soft Power collegabile all'Amico 2000. Telefonare ore pasti. Claudio Marascalchi - Via Milesi, 11 - 30174 Mestre (VE) - Tel. 041/955143

Vendiamo **programmi di ogni genere** per CBM 64, VIC 20, Spectrum, TI 99/4A. Inoltre per i computer Commodore anche cassette. Richiedere la lista a: Giuseppe Costanza - Via Cellini, 8 - 10024 Moncalieri (TO) - Tel. 011/6069872

Cambio **giochi e utility su nastro o dischetto** con medesimi. Sono molto interessato agli utility. Pierpaolo Palazzo - Via Ciccarone, 80 - 66054 Vasto (CH) - Tel. 0873/4071

Cambio, vendo **utility per proteggere e sproteggere**: Clone, Unguard, Mimic 5, Supercopy. Software gestionale per medici, ingegneri, dentisti, magazzino. RTTY, CW, Amtor, Data Base e giochi originali 1984. Paola Sbarlella - Via Valle Verde, 5 - 05100 Terni - Tel. 0744/56870

Vendo **software per tutti i mini e personal computer** a prezzi imbattibili oppure cambio con programmi per Macintosh. Invio catalogo con oltre 250 programmi. Franceso Corso - Via G. Cuomo, 53 - 84092 Bellizzi (SA)

Vendo **le prime 3 cassette di Videoteca computer** a L. 10.000. Vendo e cambio giochi in LM a prezzi scontati (F... man, Biliardo, Pit stop, Defender) velocizzati con Turbo tape. Scrivetemi. Nino Riggio - Via Esseneto, 65 - 92100 Agrigento

Vendo **cassetta con circa 40 programmi** per Sega SC 3000 con giochi, utilities, grafica 3 D, Adventure, Othello, Statistica, matematica a solo L. 25.000a comprese spese di spedizione. Carlo Ferrari - Via Merici, 90 - 00162 Roma - Tel. 06/8316783

Cambio e vendo **programmi di ogni genere**. Vendo programmi su cassetta: 8 programmi Lit. 9000 escluso prezzo della cassetta e di spedizione. Ho programmi come Pet speed, sintetizzatore vocale e altri 400. Roberto Massaron - Via Manara, 10 - 20047 Brugherio (MI) - Tel. 039/879784

Vendesi HP86B op. 009, monitor 12", scheda porta ROM, ROM progr. avanzata, ROM matrix, 5 mesi mai usato. C.D.M. - Via Lombroso, 11 - 10125 Torino - Tel. 011/687584

PERSONAL SOFTWARE LIBRI — Gruppo Editoriale Jackson

Vuoi ordinare dei libri? Spediscì questo tagliando a: Gruppo Editoriale Jackson, Via Rosellini, 12 – 20124 Milano

Nome Cognome

Indirizzo N.

Cap. Città Provincia

Partita I.V.A. (indispensabile per le aziende)

Si richiede l'emissione della fattura ☐

Inviatemi i seguenti libri:

Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

☐ Pagherò al postino il prezzo indicato + L. 3.000 per contributo fisso spese di spedizione.

☐ Allego assegno n° ..... di L. .....

☐ Non abbonato ☐ Abbonato sconto ☐ l'Elettronico ☐ Elettronica Oggi ☐ Automazione Oggi ☐ Elaktor ☐ Informatica Oggi ☐ Computerworld ☐ Bit ☐ Personal Software ☐ Strumenti Musicali ☐ Video Giochi ☐ Home computer ☐ Telecomunicazioni Oggi ☐ PC Magazine ☐ Personal O ☐ Compuscuola ☐ Super VIC ☐ Super Sync

Data ..... Firma .....

PERSONAL SOFTWARE PICCOLI ANNUNCI

Sei un lettore di Personal Software e vuoi entrare in contatto con i tuoi lettori per comprare, cambiare o vendere il tuo sistema o qualche particolare accessorio? Spediscì questo tagliando a: Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

☐ COMPRO ☐ CAMBIO ☐ VENDO

☐ Annuncio gratuito ☐ Annuncio evidenziato (allegare L. 5.000). Non si accettano fotocopie o fogli volanti allegati al coupon.

Nome ..... Cognome .....

Via ..... N. ..... C.A.P. .....

Città ..... Prov. ..... Tel. .....

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

ANNO 4 N. 24 GENNAIO 1985

DIRETTORE RESPONSABILE: Giampietro Zanga

CAPO REDATTORE: Lorenzo Barrile

DIRETTORE TECNICO: Riccardo Paolillo

COORDINAMENTO TECNICO: Luca Zaninello

GRAFICA E IMPAGINAZIONE: Ugo Luzzati

SEGRETARIA DI REDAZIONE: Silvia Borgonovo

REDAZIONE USA: Gabriella Martino, Sergio Mello-Grand
G.E.J. Publishing Group Inc.

HANNO COLLABORATO A QUESTO NUMERO:
P. Coretti, S. di Tommaso, G. Gatti,
A. Bartalesi, C. Poma, V. Delle Cave,
N. Carloni, M. Paolinelli, A. Guida,
R. Tabacco, M. Giacobazzi, M. Lenzi,
M. Spero, S. Borsani

UFFICIO ABBONAMENTI: Tel. 6880951/2/3/4/5

USA: G.E.J. Publishing Group, Inc. a Subsidiary of -
Gruppo Editoriale Jackson 1307 S. Mary Avenue -
94087 Sunnyvale (CA) Tel. (408) 7730103 - Telex: 49959972

GRAN BRETAGNA: G.E.J. Publishing L.t.d. 187 Oxford Street
London WIR1AJ Tel. (01) 4392931 - Telex: (051) 21248

PUBBLICITA': Concessionario per l'Italia e l'Estero
j Advertising S.r.l. Via Restelli, 5 - 20124 Milano
Tel. (02) 6882895-6882458-6880606 Telex: 316213 REINA I

PUBBLICITA' PER USA E CANADA:
International Media Marketing 16704 Marquardt Avenue,
P.O. Box 1217 Cerritos (CA) 90701 - (213) 926-9552

FOTOCOMPOSIZIONE: Lineacomp S.r.l.
Via Rosellini, 12 - 20124 Milano

STAMPA: Litografica S.r.l. (Busto Arsizio - MI)

Concessionario esclusivo per la diffusione
in Italia e all'Estero: SODIP - Via Zuretti, 25 - 20125 Milano

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n. 69 del 20/2/1982

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 4.000. Numero arretrato L. 8.000.
Abbonamento annuo (11 numeri) L. 34.000; per l'Estero L. 51.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano - mediante emissione
di assegno bancario, cartolina vaglia o utilizzando
il C/C postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al
nuovo, anche l'indirizzo precedente, ed allegare alla
comunicazione l'importo di L. 500, anche in francobolli.

© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

Il Gruppo Editoriale Jackson è iscritto nel Registro nazionale
della Stampa al n. 117 - Vol. 2 - Foglio 129 in data 17-8-1982

GRUPPO EDITORIALE JACKSON S.r.l.

DIREZIONE, REDAZIONE, AMMINISTRAZIONE:
Via Rosellini, 12 - 20124 Milano - Telefoni: 68.80.951/2/3/4/5

SEDE LEGALE: Via G. Pozzone, 5 - 20121 Milano

DIREZIONE EDITORIALE: Daniele Comboni

DIREZIONE AMMINISTRATIVA: Giuliano Di Chiano

DIREZIONE DIVISIONE PERIODICI: Dario Tiengo

DIREZIONE DIVISIONE LIBRI
E GRANDI OPERE: Roberto Pancaldi

Libri firmati JACKSON

Massimo Mangia
OLIVETTI M10: GUIDA ALL'USO
Una guida all'uso, ma anche una precisa fonte di idee e di possibili applicazioni.
Il libro è diviso in 2 sezioni: nella prima sono descritti i comandi e le istruzioni del linguaggio BASIC, classificati in gruppi funzionali, con un criterio che ne semplifica l'apprendimento e la consultazione.
Nella seconda parte vengono presentati i programmi applicativi integrati nel calcolatore, che ne fanno di volta in volta una versatile macchina da scrivere, un'agenda, un indirizzario o un terminale di un sistema remoto.
192 pagine
Codice **401B L. 18.000**

Rita Bonelli - Daria Gianni
M20 LA PROGRAMMAZIONE BASIC-PCOS
Un libro completo sul personal italiano più famoso; Per la lettura è richiesta la conoscenza di alcuni concetti elementari di informatica di base e dei sistemi di numerazione binario, ottale ed esadecimale.
La presentazione sistematica dei comandi PCOS e delle istruzioni BASIC è accompagnata da una ricca gamma di esempi e applicazioni, che portano gradualmente il lettore a conoscere il sistema operativo e le tecniche di programmazione, dalle più semplici a quelle più sofisticate.
360 pagine
Codice **401A L. 30.000**

Rita Bonelli
IL PRIMO LIBRO PER M24 MS DOS E GW BASIC
Il libro offre una panoramica rapida ma completa sul nuovo Personal computer M24.
Il primo capitolo, dopo aver posto l'accento sul binomio calcolatore-pacchetti di programmi, che rappresenterà sempre di più la carta vincente nell'evoluzione del mercato, descrive la configurazione hardware del sistema M24, che fa da supporto a diversi sistemi operativi e diversi linguaggi.
Segue un capitolo che illustra l'utilizzo della macchina nei diversi campi e le prove di alcuni pacchetti di software disponibili.
Gli ultimi due capitoli descrivono uno dei sistemi operativi, l'MS-DOS, e uno dei linguaggi, il GW-BASIC, riportando alcuni esempi.
152 pagine
Codice **401P L. 24.000**

La Biblioteca che fa testo

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
		Totale	

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca _____ n° _____

☐ Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

☐ 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

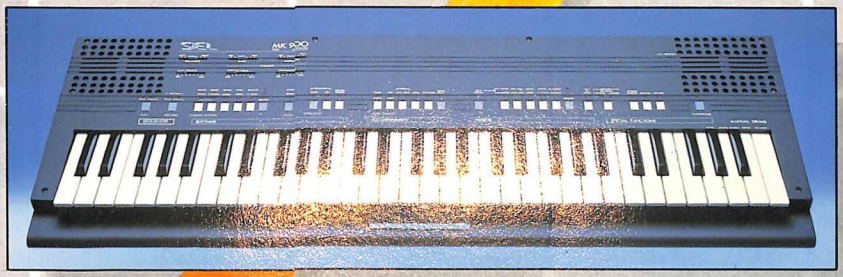
GRUPPO EDITORIALE JACKSON

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

nuovidea

INCREDIBILE

TASTIERA - MIDI - SEQUENCER - BATTERIA PROGRAMMABILE COMPUTER COMPATIBILE...



MK900

MIDI KEYBOARD

Tastiera portatile stereo - MIDI compatibile - a doppia generazione sonora

Possibilità di collegamento a computers

Supporto stand ripiegabile (opzionale)

Una straordinaria ricchezza timbrica e una insuperabile versatilità sintetizzate in uno strumento a microprocessore dalla estrema facilità d'uso

Pedale d'espressione (opzionale)

MIDI IN e MIDI OUT

10 ritmi + 1 ritmo programmabile dall'utente

10 Presets a doppia generazione sonora

Sequencer in tempo reale: 260 note + pause, 50 accordi, batteria per memorizzazione dati

Midi Computer Interface (opzionale)

Divisione della tastiera programmabile che permette di suonare contemporaneamente 2 timbri oppure un solo timbro con polifonia 14

Demo Song

Accompagnamento automatico multifunzione

Controcanto automatico

Transpose, Detune, Stereo Chorus

Amplificazione stereo con due altoparlanti biconici a sospensione pneumatica incorporati

Tastiera a 61 tasti

Peso: kg. 6

SIEL®

Distribuito da

ARAMINI

STRUMENTI MUSICALI

Cadriano di Granarolo, via B. Buozzi, 1b (Bologna)

Tel. 051/766.077