

SUPER

L. 7.500

Aprile 1986 Anno 1 n.1

COMMODORE **16** **plus/4**

IL TRASFERIMENTO DEI FILE SU DISCO

COME UTILIZZARE L'INTERFACCIA RS-232

SCRITTE GIGANTI

HOUSE OF TERROR PER VIC 20

ZAP! PER C16-PLUS/4

ASTEROIDI PER C16-PLUS/4

LA TANA DEL DRAGONE

GUIDA PERICOLOSA PER C16-PLUS/4

BATTAGLIA NAVALE

**SUPERCALC: UN FOGLIO
ELETTRONICO PER C16**

SEQFILING SYSTEM 1.0 PER VIC 20

Spedizione in abbonamento postale Gruppo III/70



**GRUPPO EDITORIALE
JACKSON**

DIVISIONE PERIODICI



**GRUPPO EDITORIALE
JACKSON**

DIVISIONE LIBRI



Scopri il tuo home computer

A.P. Stephenson
**IL COMPUTER:
l'Hobby e il lavoro**

Il computer come compagno
ideale, nella propria
professione e nell'hobby.
Cod. CH187 Pag. 192 L. 12.000

*R. Bonelli, L. Pazzucconi,
F. Racchi*
**GUIDA AL COMMODORE
PLUS 4**

Un manuale completo per il
Plus 4 ripercorrendo C16 per
te e C16 sempre di più.
Cod. CC256 Pag.470 L. 30.000

G. Marano
**30 PROGRAMMI
PER MSX**

Inserite soltanto la cassetta
e... una valanga
di programmi uscirà
dal vostro MSX.
Cod. CZ181 Pag. 96 L. 20.000
Con cassetta

A. Cattaneo, P. Todorovich
COMPUTER HARDWARE

Il primo vero libro di
realizzazioni pratiche
dedicate ai più diffusi home
computer: C64, Spectrum,
VIC20, C16 e MSX.
Cod. CH182 Pag. 152 L. 13.000

A. Borra, M.C. Grizzi
**LAVORIAMO CON
IL COMMODORE 16**

Lavorare con il C16 è un
piacere, fallo anche tu,
ti divertirai.
Cod. CC244 Pag. 196 L. 20.000
Con cassetta

Jackson
la biblioteca che fa testo

COMMODORE 16 plus/4

4 
LA POSTA DEI LETTORI

25 
25 HOUSE OF TERROR
PER VIC 20
di **M. Marinai**

47 SUPERCALC: UN
FOGLIO ELETTRONICO
PER C16
di **M. Tinglof**
Trad. e adatt.
di **M. Cristuib Grizzi**

9 
IL TRASFERIMENTO
DEI FILE SU DISCO
di **E. Sarcina**

31 SCRITTE GIGANTI
di **D. Smoak**
Trad. e adatt.
di **F. Sarcina**

52 ZAP!
PER C16-PLUS/4
di **T. Marazita**
Trad. e adatt.
di **A. Borra**

18 
COME PROTEGGERE
DALLA SCRITTURA I
PROPRI DISCHETTI
di **K. Hawkins**
Trad. e adatt.
di **S. Albarelli**

34 ASTEROIDI
PER C16-PLUS/4
di **A. Motta**

57 GUIDA PERICOLOSA
PER C16-PLUS/4
di **A. Borra**

37 SEQFILING
SYSTEM 1.0
PER VIC 20
di **M. Cristuib Grizzi**

60 LA TANA DEL
DRAGONE
di **C. Book**
Trad. e adatt.
di **F. Sarcina**

21 
COME UTILIZZARE
L'INTERFACCIA RS-232
di **M. Cristuib Grizzi**

43 BATTAGLIA NAVALE
di **J. Butterfield**
Trad. e adatt.
di **R. Luzzo**

64 
PICCOLI ANNUNCI

65 
RECENSIONI LIBRI

GRUPPO EDITORIALE JACKSON
s.r.l.
San Francisco - Londra - Milano

**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Via Rosellini, 12
20124 MILANO
Tel. (02) 68.80.951/2/3/4/5
Telex 333436 GEJIT I

SEDE LEGALE
Via G. Pozzone, 5 - 20121 Milano

CONSOciate ESTERE
USA
GEJ Publishing Group, Inc.
a subsidiary of
Gruppo Editoriale Jackson
1307 S. Mary Avenue - 94087 Sunnyvale CA
Tel. (408) 7730138 - Telex 49959972

GRAN BRETAGNA
GEJ Publishing Ltd.
187 Oxford Street London WURIAJ
Tel. (01) 4392931 - Telex (051)21248

DIREZIONE EDITORIALE
Daniele Comboni

DIREZIONE AMMINISTRATIVA
Giuliano Di Chiano

**DIREZIONE DIVISIONE LIBRI E
GRANDI OPERE**

Roberto Pancaldi
DIRETTORE RESPONSABILE
Giampietro Zanga
REDAZIONE
Luca Zaninello
**HANNO COLLABORATO A
QUESTO NUMERO**

S. Albarelli
A. Borra
M. Cristuib Grizzi
R. Luzzo
M. Marinai
A. Motta
E. Sarcina
F. Sarcina

GRAFICA E IMPAGINAZIONE
Ivana Rossi
FOTOCOMPOSIZIONE

Graphotek
Via Astesani, 16 - 20161 Milano
STAMPA
Centro Poligrafico Milan s.r.l.
20089 Ponte Sesto di Rozzano (MI)

PUBBLICITÀ
Concessionaria per l'Italia e l'Estero
J.Advertising s.r.l.

Viale Restelli, 5 - 20124 Milano
Tel. (02) 68.82.895-68.82.48-68.80.606
TELEX 316213 REINA I
**CONCESSIONARIO PUBBLICITÀ PER
USA E CANADA**
International Media Marketing
16704 Marguardt Avenue
P.O. Box 1217 - Cerritos CA 90701
(213) 926 - 9552

DISTRIBUZIONE
SODIP
Via Zuretti, 25 - 20125 Milano
Spedizione in abbonamento postale
Gruppo III/70
Pubblicità inferiore al 70%
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento L. 82.500
(11 numeri con cassetta);
L. 38.500 (11 numeri senza cassetta)
per l'Estero L. 110.000 (11 numeri con
cassetta); L. 50.000 (11 numeri senza cassetta)

I versamenti vanno indirizzati a:
Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano
mediante emissione di assegno bancario,
vaglia o utilizzando il C.C.P. n° 11666203

Per i cambi di indirizzo indicare, oltre al
nuovo, anche l'indirizzo precedente
e allegare L. 500, anche in francobolli.

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE**
Tribunale di Milano in atto

GLI ARTICOLI TRADOTTI SONO TRATTI
DALLE RIVISTE
COMPUTE! O DA
COMPUTE! GAZETTE
COMPUTE! PUBLICATION, INC.
P.O. Box 5406, Greensboro, NC 27403 - USA

**Il Gruppo Editoriale Jackson è
iscritto nel Registro nazionale della
Stampa al n° 117 vol. 2 - foglio 129
in data 17-8-1982**

Rivista associata
all'Unione
Stampa
Periodica
Italiana



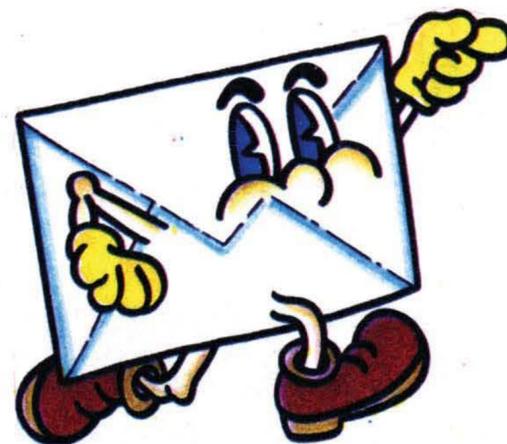
ASSOFT



Associata

READ & WRITE

La posta
dei lettori



La lunghezza dei programmi

Volendo ordinare in maniera organica i miei programmi su cassetta, vorrei indicare accanto al nome di ogni programma la sua lunghezza in byte. Dato che la funzione FRE mi fornisce a volte valori addirittura negativi, credo che non sia molto affidabile. Potete indicare un metodo diverso per conoscere l'esatta occupazione di memoria di un programma?

P. Boncompagni

R. Effettivamente il valore fornito dalla funzione FRE(0) è tutt'altro che preciso: per avere il valore esatto della memoria occupata da un programma, sia con il C16 e Plus/4 che con il VIC 20, bisogna invece digitare:

```
PRINT2+FRE(0)-(FRE(0)<0)*65536
```

Il valore che verrà visualizzato tiene anche conto dello spazio in memoria eventualmente occupato da variabili, matrici e vettori.

Se si desidera invece conoscere la quantità di memoria occupata unicamente dalle istruzioni di un programma, al netto cioè di variabili, matrici e vettori, è necessario ricorrere ai puntatori nelle locazioni 43, 44, 45 e 46. Le locazioni 43 e 44 contengono (nella forma byte basso-byte alto) la locazione più bassa usata dal BASIC, mentre le locazioni 45 e 46 contengono (nello stesso formato) la locazione di inizio delle variabili. L'esatta quantità di memoria occupata dalle istruzioni di un programma (cioè dal "listato") è fornita da:

```
PRINT(PEEK(45)+PEEK(46)*256)-  
(PEEK(43)+PEEK(44)*256)-2
```

Il numero 2 in entrambe le istruzioni serve per compensare la quantità di memoria, appunto di 2 byte, occupata dall'istruzione stessa.

Speedscript per C16?

Avete pubblicato sul numero di Ottobre 1985 due utilissimi programmi per VIC 20, cioè "Speedscript 3.0" e "Relfiling System 1.0": perché non ne avete fatto anche una versione per C16, mentre l'avete fatta per il C64? Dal momento che non esistono in commercio altri programmi "seri di word processing e di archivio per il C16, speravo che avreste potuto sopperire voi stessi a questa carenza nel mercato del software per questo computer.

P. Bonafede

R. Le versioni per C16 di questi programmi in realtà esistono, ma essendo comprese nel libro con cassetta "Lavoriamo con il Commodore 16", pubblicato dal Gruppo Editoriale Jackson, ovviamente non ci è possibile pubblicarle sulla rivista. Il libro cerca appunto di sopperire alla carenza di software "serio" per il C16, e comprende un word-processor in linguaggio macchina - "EasyWord" - dotato delle stesse prestazioni di SpeedScript 3.0, oltre a due database - "Relfiling System" e "Seqfiling System" - e a uno spreadsheet - "MultiCalc" - inedito anche per i lettori "sessantaquattristi" della nostra rivista.

Trucchetti con il drive

Se cerco di caricare da disco un file SEQ, ottengo sempre un messaggio "FILE NOT FOUND", anche se il file è lì visibilissimo sulla directory. È il drive che non funziona o sono io che sbaglio qualcosa nel caricamento?

G. Fossati

R. I file sequenziali (SEQ) non si possono caricare come i programmi (PRG), ma richiedono una particolare procedura per essere letti dal computer. Questo

programmino visualizza su video il contenuto di qualsiasi file sequenziale presente sul disco:

```
10 INPUT"NOME FILE SEQ";F$  
20 OPEN3,8,3,F$  
30 GET#3,A$:PRINTA$;  
40 IF ST=64THENCLOSE3:END  
50 GOTO30
```

Una particolarità interessante e poco conosciuta nel formato delle istruzioni LOAD e SAVE permette a questo proposito di realizzare una forma elementare di protezione dei propri programmi su disco. Se infatti si salva un programma con SAVE"TEST,S",8 esso apparirà sulla directory come SEQ, ed ogni tentativo di caricarlo con LOAD"TEST",8 darà invariabilmente il messaggio "FILE NOT FOUND". Se invece si usa la forma LOAD"TEST,S",8 il programma verrà normalmente caricato in memoria e potrà essere mandato in esecuzione.

Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERCOMMODO-**
RE contengono particolari caratteri di controllo
(controllo cursore, tasti colore, video-inverse, ecc.).

Per rendere chiaro ciò che dev'essere battuto quando
è necessario inserire uno dei suddetti caratteri sono
state stabilite le seguenti convenzioni.

Generalmente i listati contengono delle «parole» racchiuse
tra parentesi graffe { }; tali parole rappresentano
particolari caratteri di controllo: {GIÙ} significa premere
il simbolo del cursore verso il basso. {5 SPAZI} vuol dire
battere 5 volte la barra-spazio.

Per indicare che un tasto dev'essere «shiftato», cioè
premutato insieme al tasto SHIFT, nel listato il simbolo
di questo tasto sarà sottolineato.

Per esempio S significa che dev'essere premuto il tasto
S mentre è abbassato il tasto SHIFT.

La manovra farà apparire sullo schermo un piccolo
«cuore».

Trovando un simbolo sottolineato chiuso tra parentesi
graffe (esempio {10 N}) dovrà essere interpretato come
«premere il carattere indicato per il numero di volte
che lo precede nelle parentesi»: nel nostro esempio
premere 10 volte il tasto N «shiftato».

Se il tasto da premere è indicato tra le parentesi [>],
vuol dire che lo stesso dovrà essere premuto mentre
è premuto il tasto «Commodore» (il tasto «Commodo-
re» è quello posto nell'angolo in basso a sinistra).

Se il tasto scritto tra [>] è preceduto da un numero,
ciò significa che il tasto dev'essere premuto per il numero
di volte indicato.

Raramente si potrà trovare un carattere alfabetico
solitario racchiuso tra parentesi graffe. Tale carattere
può essere battuto mentre è premuto il tasto
CTRL.

Ad esempio {A} sta ad indicare la sequenza CTRL-A.
A proposito del «modo virgolette» è nota la possibilità
di muovere il cursore sullo schermo con i tasti CRSR.
Ogni tanto i programmatori desiderano muovere il cursore
durante l'esecuzione del programma. È per questo motivo
che nei programmi si troveranno dei {SIN},
{HOME} e {BLU}.

L'unico modo perché il calcolatore distingua il comando
diretto da quello inserito nel programma è il «modo
virgolette». Premendo il tasto «virgolette» (tasti SHIFT
2) il calcolatore si predispone appunto in «modo virgo-
lette».

Battendo un carattere qualsiasi, cercando poi di cor-
reggerlo muovendo il cursore, si otterrà solo un tratto
verticale in un quadratino inverso. Questo è infatti
il simbolo del cursore a sinistra nel «modo virgolette».
L'unico comando di editing non utilizzabile all'interno
di un programma è il DEL. Battendo nuovamente le
«virgolette» il calcolatore lascia il «modo virgolette». Si
può accedere al «modo virgolette» quando inserite spa-
zi nella linea.

In ogni caso, il sistema più semplice per uscire dal
«modo virgolette» è quello di premere il tasto RE-
TURN.

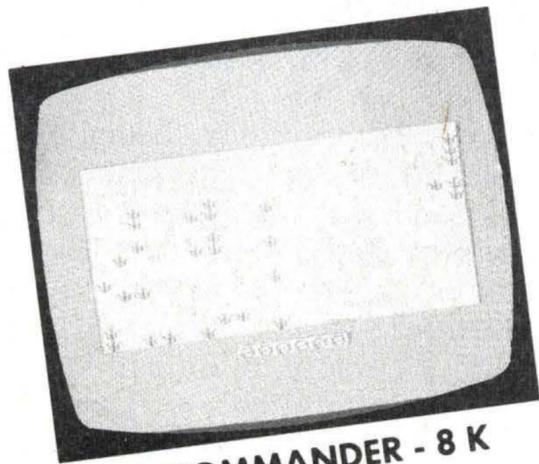
Utilizzare la tabella che segue quando è necessario
inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	G 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	G 8	
{SU}	SHIFT ↑ CRSR ↓		{GRN}	CTRL 6		{F1}	F1	
{GIU'}	↓ CRSR ↓		{BLU}	CTRL 7		{F2}	F2	
{SIN}	SHIFT ← CRSR →		{YEL}	CTRL 8		{F3}	F3	
{DES}	← CRSR →		[<1>]	G 1		{F4}	F4	
{RVS}	CTRL 9		[<2>]	G 2		{F5}	F5	
{OFF}	CTRL 0		[<3>]	G 3		{F6}	F6	
{BLK}	CTRL 1		[<4>]	G 4		{F7}	F7	
{WHT}	CTRL 2		[<5>]	G 5		{F8}	F8	
{RED}	CTRL 3		[<6>]	G 6				

SOFTWARE SELEZIONATO PER IL TUO VIC-20

Una selezione di giochi recenti scelti tra i migliori disponibili sul mercato. Ciò che aspettavate per divertirvi.

L. 8.000

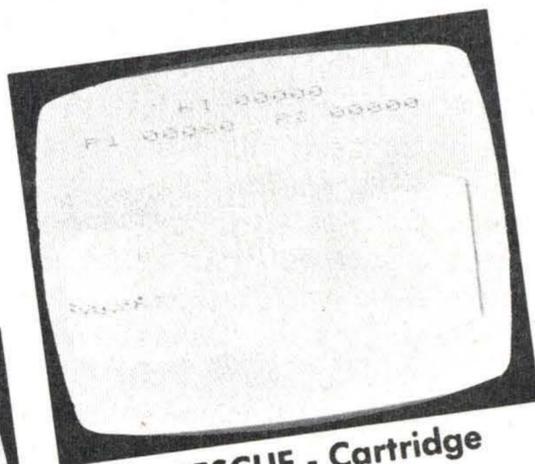
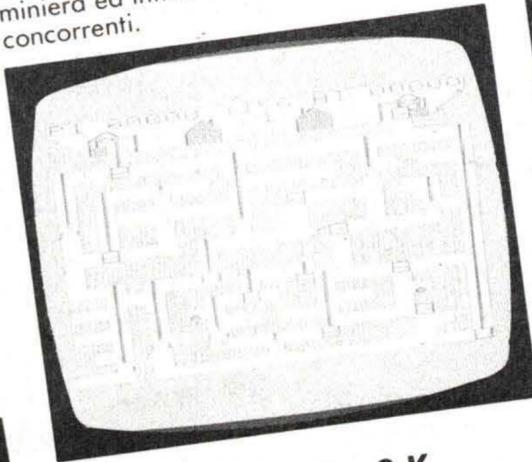


TANK COMMANDER - 8 K

COD. DTEMV03
Alla guida del proprio carro armato lungo un'area accidentata, con fiumi ed ostacoli. Improvvisamente appaiono i carri nemici che si debbono eliminare prima di essere distrutti.

MINE MADNESS - Cartridge

COD. DTEMV06
Esploratori in cerca d'oro si avventurano in una miniera trovandosi alle prese con enormi montacarichi per salire e scendere, con mostri che popolano la miniera ed infine con altri esploratori concorrenti.



RIVER RESCUE - Cartridge

COD. DTEMV05
Spedizione lungo un fiume al salvataggio di alcuni scienziati dispersi, evitando tronchi d'albero, coccodrilli e banchi di sabbia. Si tratta di un classico arcade d'azione.

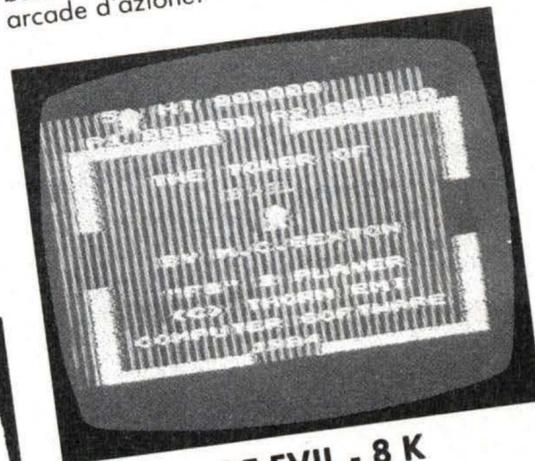
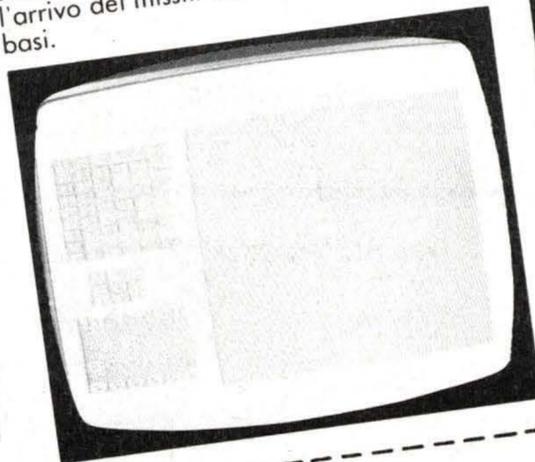


SUBMARINE COMMANDER - 16 K

COD. DTEMV01
Una realistica simulazione di guerra navale nel Mediterraneo, nella quale ci si trova al comando di un sommergibile. Varie e diverse schermate permettono di controllare assetto e proprie posizioni per raggiungere ed affondare i convogli nemici.

COMPUTER WAR - 8 K

COD. DTEMV02
Simulazione di attacco nucleare che si ispira al film "War Games". Permette di salvare il destino del mondo bloccando l'arrivo dei missili nemici verso le proprie basi.



TOWER OF EVIL - 8 K

COD. DTEMS02
Viaggio nelle stanze di un castello popolato di pericolosissimi mostri alla conquista di tesori e oggetti di varia utilità. Un gioco impegnativo e coinvolgente, adatto anche a coloro che non amano gli arcade.

APPROFITTA!

Ritaglia ed invia a **J.soft**
Viale Restelli, 5
20124 Milano

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

Ordino il seguente software per il VIC 20

cod. _____ cod. _____ cod. _____
cod. _____ cod. _____ cod. _____
cod. _____ cod. _____ cod. _____

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

Scelgo la seguente modalità di pagamento: pagherò in contrassegno al postino
 allego assegno (o contanti)
 verso l'importo sul C.C.P. n. 19445204 intestato a J.soft (allego ricevuta)

Nome _____
Cognome _____
Via _____ n. _____
C.A.P. _____ Città _____ Prov. _____

OK-LIST

Sappiamo tutti quanto sia difficile digitare correttamente un programma (ovviamente, se non si dispone della cassetta allegata a SUPERCOMMODORE!). Errori di battitura anche banali possono provocare il malfunzionamento del programma stesso, o anche il blocco completo del sistema, obbligandoci a rifare il lavoro.

Ma il punto di forza dei calcolatori è che essi eccellono nei compiti noiosi e che richiedono precisione. Scoprire un errore nella battitura di un listato è appunto un compito noioso, quindi perché non sfruttare il tuo computer per controllare i listati da te digitati?

È proprio questo lo scopo di OK-LIST, il programma che ti permetterà di risparmiare tempo e fatica. Annidato all'interno del tuo computer, OK-LIST controlla automaticamente ogni riga battuta, mostrandoti un numero in cima allo schermo ogni volta che premi il tasto RETURN. Un numero corrispondente nel listato del programma pubblicato ti permette il confronto, ed uno sguardo è sufficiente per avere la conferma di aver digitato la linea correttamente.

OK-LIST è un breve programma in linguaggio macchina residente nel buffer del registratore. Esso rimane in tale zona finché spegni il computer o usi l'unità a cassette.

Come digitare OK-LIST

Digita il listato che segue, facendo at-

tenzione a non commettere errori (per fortuna il listato è molto breve!). Dopo aver finito salvato su disco o nastro, per lo meno un paio di volte, prima di mandarlo in esecuzione con il comando RUN, perché il programma BASIC, dopo aver memorizzato la routine in linguaggio macchina (le istruzioni DATA), si autocancella.

Come si usa OK-LIST

A questo punto puoi iniziare a battere il tuo primo listato "a prova di errore". Se non hai OK-LIST in memoria, caricalo e attivalo con RUN.

Digita quindi la prima linea di programma: come puoi vedere, in fondo ad essa c'è la parola ":rem" seguita da un numero, che si chiama checksum.

NON COPIARE QUESTA PARTE DELLA LINEA! Ora premi il tasto RETURN, come al solito. In alto a sinistra apparirà il checksum; confrontalo con quello che vedi scritto sulla rivista dopo ":rem". Se i due numeri sono uguali, vuol dire che la linea è stata digitata correttamente, altrimenti controllala, perché evidentemente hai commesso un errore. Facile, no?

OK-LIST non è pignolo per quel che riguarda gli spazi bianchi, dato che non fa caso alla loro presenza. Ciò torna a tuo vantaggio, dal momento che la spaziatura non è, in genere, importante. Ma all'interno delle istruzioni PRINT gli spazi spesso si rivelano necessari, quindi presta attenzione quando digiti questo tipo di istruzione. Un'altra cosa a cui devi badare è che usando le forme abbreviate dei comandi (ad esempio ? per PRINT) il checksum visualizzato

non coinciderà con quello presente sul listato. Ciò è tuttavia risolvibile, premendo il tasto RETURN su una linea dopo averla listata: le abbreviazioni saranno ricodificate nella loro forma intera e quindi OK-LIST sarà in grado di svolgere perfettamente la sua funzione. In questo modo, inoltre, avrai la possibilità di controllare interi gruppi di righe in una volta sola.

Ovviamente, nessuno è perfetto, ed anche OK-LIST ha qualche limite: se per caso ti capitasse (ma è un errore poco frequente) di "anagrammare" durante la digitazione una linea (per esempio invece di battere 10 PRINT "VAI", digiti per errore 10 PRINT "VIA"), OK-LIST non se ne accorgerebbe. Inoltre la presenza del carattere SHIFT-SPAZIO modifica il numero di checksum rispetto al carattere SPAZIO normale, anche se la differenza non risulta visibile sul video del tuo computer.

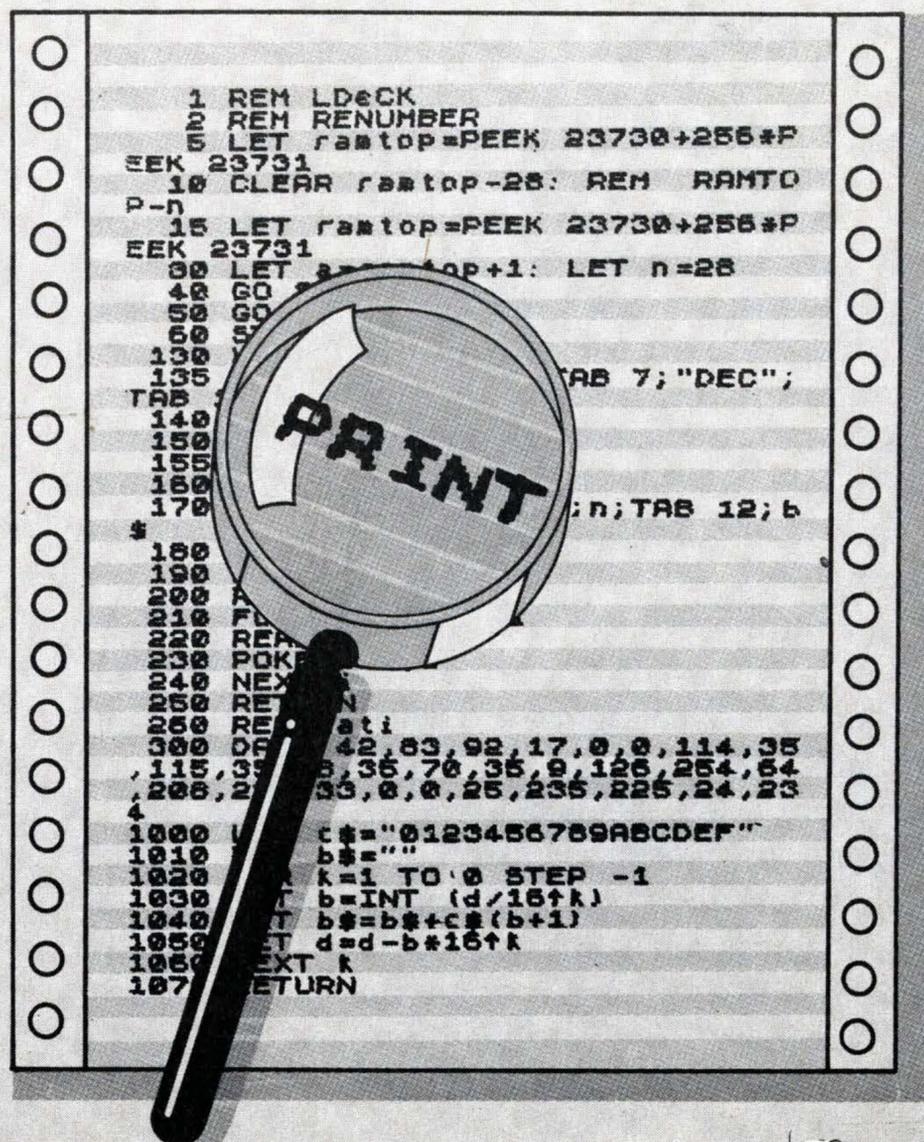
Se intendi sospendere il lavoro di digitazione - o lo hai terminato - puoi salvare su nastro o disco il tuo programma: ricordati comunque di premere RUN/STOP e RESTORE prima di dare il comando SAVE. Al momento di riprendere il lavoro, tuttavia, non potrai più utilizzare OK-LIST, dal momento che qualsiasi operazione su cassetta lo distrugge.

Se vuoi disabilitare OK-LIST in qualunque momento, senza perdere il tuo programma in memoria, premi RUN/STOP e RESTORE. Per riattivarlo bat-

SYS 886

Buon lavoro!

```
10 PRINT "{CLR}ATTENDERE PREGO..."
11 FORI=886TO1018:READA:CK=CK+A:POKEI,A:NEXT
12 IF CK<>17539 THEN PRINT "{GIU'}ERRORE NELLE ISTRUZIONI DATA":END
13 SYS886:PRINT "{CLR}{ 2 GIU'}OK-LIST ATTIVATO.":NEW
14 DATA173,036,003,201,150,208
15 DATA001,096,141,151,003,173
16 DATA037,003,141,152,003,169
17 DATA150,141,036,003,169,003
18 DATA141,037,003,169,000,133
19 DATA254,096,032,087,241,133
20 DATA251,134,252,132,253,008
21 DATA201,013,240,017,201,032
22 DATA240,005,024,101,254,133
23 DATA254,165,251,166,252,164
24 DATA253,040,096,169,013,032
25 DATA210,255,165,214,141,251
26 DATA003,206,251,003,169,000
27 DATA133,216,169,019,032,210
28 DATA255,169,018,032,210,255
29 DATA169,058,032,210,255,166
30 DATA254,169,000,133,254,172
31 DATA151,003,192,087,208,006
32 DATA032,205,189,076,235,003
33 DATA032,205,221,169,032,032
34 DATA210,255,032,210,255,173
35 DATA251,003,133,214,076,173
36 DATA003
```



Per collaborare a SUPERCOMMODORE

La rivista è interessata ad articoli e programmi per C64 e C128 di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Sprite Ed.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente.

È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come sug-

gerimenti di programmazione si consiglia di usare le istruzioni "CHR\$(x)", "TAB(x)", "SPC(x)", piuttosto che stringhe di manipolazione del cursore.

Ad esempio, per uno scroll di 5 linee l'istruzione "FORI = 1TO5:PRINT:NEXT". è molto più interpretabile di 5 Q inverse; e, invece di una dozzina di simboli di cursore a destra, perché non usare semplicemente "PRINT SPC (12)"? Un rapido controllo dei programmi per operare queste sostituzioni sarà molto apprezzato da noi e dai lettori.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24x36, o 6x6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 verrà data comunicazione scritta solo in caso di accettazione del lavoro. Il materiale non pubblicato non verrà restituito.

15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviare idee e programmi a:
SUPERCOMMODORE
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione



OPEN

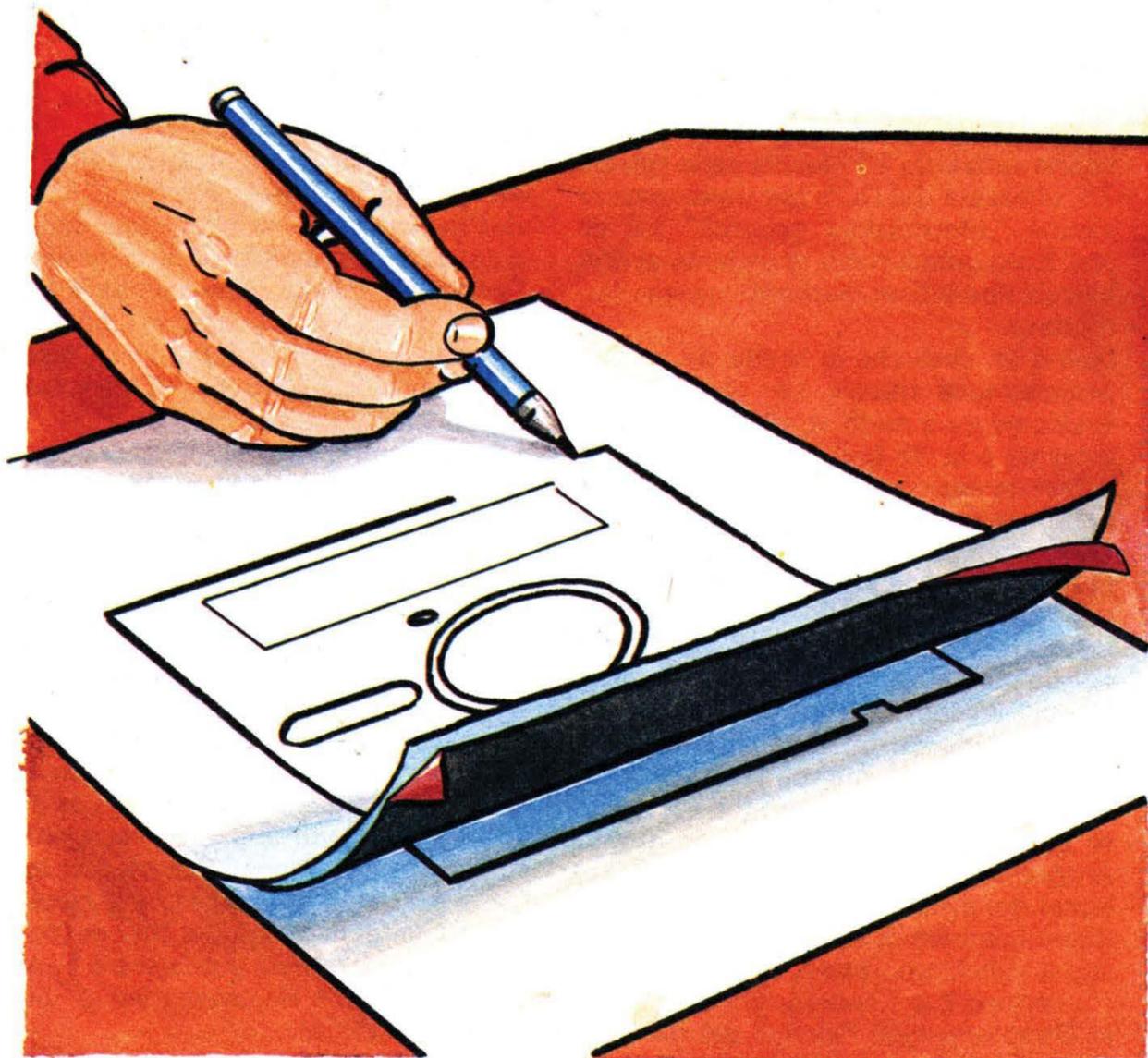
Il trasferimento dei file su disco

di **E. Sarcina**

Molti lettori sicuramente non hanno le idee chiare sul modo in cui vengono registrati programmi e dati su disco. In questo articolo cerchiamo di dare una risposta alle domande più ricorrenti e forniamo un utile programma in grado di trasferire i file da un disco all'altro.

Probabilmente tutti voi avete in casa un giradischi, e anche i pochi che non ne possiedono uno hanno sicuramente avuto occasione di utilizzare questo apparecchio. Esso è in grado di leggere un segnale sonoro da un disco che gira, tramite una testina libera di spostarsi lungo il suo raggio; inoltre la lettura avviene con uguale facilità (e praticamente con lo stesso "tempo di accesso") qualunque sia il punto del disco che contiene il pezzo desiderato. Quando invece vogliamo ascoltare un brano registrato su una cassetta, dobbiamo far avanzare o riavvolgere il nastro fino al punto in cui si trova la registrazione; ciò significa che il tempo di accesso è notevolmente variabile e può diventare anche molto lungo.

Questa piccola premessa serve a introdurre un'analogia tra il giradischi, strumento familiare, e il disk drive, che per molti resta un oggetto misterioso, una "scatola nera" il cui funzionamento è ignoto. Proprio come il giradischi, il drive legge le informazioni da appositi dischi (e in più può anche scriverle). La testina, per fortuna, non deve venir spostata a mano: il suo movimento è comandato direttamente da un microprocessore, un "cervello" analogo a quello del computer, che esegue le istruzioni



di un programma chiamato DOS (Disk Operating System = Sistema Operativo del Disco). Proseguendo le analogie con l'esempio musicale, nel disk drive l'accesso alle informazioni è "casuale", cioè non dipende dalla loro posizione sul disco; in una unità a nastro, invece,

l'accesso è "sequenziale", in quanto per trovare un'informazione dobbiamo leggere in sequenza tutte le precedenti.

Tracce e settori

Ma come sono disposte le informazioni su un floppy disk? Anche qui ci viene

in aiuto il vecchio giradischi. Così come un LP è solitamente diviso in zone concentriche separate, ognuna delle quali contiene un brano musicale, anche un dischetto magnetico viene suddiviso in un certo numero di tracce concentriche (35 nel caso del Commodore 1541) di uguale larghezza. In questo caso, però, la divisione non è visibile, in quanto si tratta di una registrazione magnetica e non di un'incisione fisica. La testina, spostandosi lungo il raggio del floppy, accede a una delle tracce; per convenzione la traccia n.1 è quella più esterna e la n.35 è quella più vicina al centro del dischetto. Ogni traccia viene poi a sua volta suddivisa in un numero di settori dipendente dalla sua lunghezza (ovviamente, le tracce esterne sono più lunghe di quelle interne: esse hanno una circonferenza maggiore trovandosi più lontane dal centro). I settori sono numerati a partire da zero e disposti come segue:

tracce 1-17: 21 settori (0-20)
 tracce 18-24: 19 settori (0-18)
 tracce 25-30: 18 settori (0-17)
 tracce 31-35: 17 settori (0-16)

La suddivisione del disco in tracce e settori avviene al momento della formattazione. Ogni settore contiene un blocco di informazioni (256 byte). Il numero totale di blocchi è quindi $(21 \times 17) + (19 \times 7) + (18 \times 6) + (17 \times 5) = 683$, da cui vanno sottratti i 19 blocchi della traccia 18, utilizzata dal DOS per memorizzare alcune "informazioni generali" sul disco. Ecco quindi come giungiamo ai 664 blocchi liberi di un dischetto appena formattato.

Non è un caso che la scelta sia caduta proprio sulla traccia 18, che si trova esattamente a metà strada tra la 1 (più esterna) e la 35 (più interna): dato che la testina accede ai vari file sempre partendo da questa traccia, il tempo d'accesso viene minimizzato. Le informazioni contenute nella traccia 18 sono le seguenti:

- Il settore 0 contiene la BAM (Block Availability Map = mappa di disponibilità dei blocchi), un'area di 140 byte in cui a ogni blocco del disco viene associato un bit. Questa mappa indica al DOS se un certo blocco è libero (bit=1) oppure occupato (bit=0). Sempre in questo settore è memorizzata la "Header" (Testata) del dischetto, cioè il nome, e l'ID (Identificatore) di due caratteri che noi stessi indichiamo per formattarlo.

- Il settore 1 e i successivi contengono la Directory, cioè l'elenco dei file registrati sul disco. Ogni blocco contiene le informazioni relative a 8 file. Per ogni file viene indicato il tipo (programma, sequenziale, ecc.), il blocco iniziale a partire da cui è registrato, il nome (16 caratteri) e la lunghezza in blocchi. Un dischetto può contenere quindi un massimo di $18 \times 8 = 144$ file.

Quando registriamo un file sul disco è il DOS che si occupa di aggiornare la Directory e la BAM, "allocando" i bloc-

chi utilizzati, cioè facendoli risultare occupati nella mappa, in modo che le informazioni non possano in seguito venir cancellate. La scelta dei blocchi da utilizzare avviene esplorando il disco a partire dalla traccia 18 e procedendo sia verso l'interno che verso l'esterno; cioè, in parole povere, prima vengono occupate le zone "centrali", cioè vicine alla traccia 18, e poi man mano si avvanza di pari passo verso le due "periferie", interna ed esterna. In tal modo le ultime tracce ad essere utilizzate sono la 35 e la 1.

I file registrati su disco possono essere di quattro tipi: PRG (programma), SEQ (sequenziale),USR (utente) e REL (relativo). I primi tre tipi vengono trattati in modo analogo dal DOS (si può anzi dire che gli USR sono del tutto identici ai SEQ), mentre i file relativi sono strutturati in modo diverso e più complesso, ed esulano dal tema di questo articolo. Ci occuperemo quindi soltanto dei file-programma e dei sequenziali, che differiscono tra di loro sostanzialmente per il fatto che solo i primi possono essere registrati con un comando SAVE e letti con un LOAD. Un file PRG è la registrazione di un'area di memoria del computer; i primi due byte del file contengono l'indirizzo iniziale dell'area salvata. Se in seguito per leggere il file si digita LOAD "nome",8 il programma viene caricato a partire dall'inizio della RAM BASIC, mentre se si digita LOAD "nome",8,1 il file viene caricato a partire dall'indirizzo specificato nei due byte di cui sopra, che può benissimo non essere l'inizio del BASIC (specialmente se si tratta di un programma in linguaggio macchina). Un file SEQ, invece, è semplicemente una sequenza di informazioni (numeri, stringhe, ecc.).

I programmi e i file sequenziali occupano il numero di blocchi specificato nella Directory in cui è indicato anche l'indirizzo (traccia e settore) del primo blocco di informazione. I byte di ogni blocco sono numerati da 0 a 255, ma solo 254 di essi sono disponibili per l'informazione, in quanto i primi due byte sono riservati. Per l'esattezza, in tutti i blocchi, tranne l'ultimo, i byte 0 e 1 contengono traccia e settore relativi al blocco successivo, costituendo in tal modo un puntatore, e i byte 2-255 contengono il "pezzo" di file. Nell'ultimo blocco, invece, il primo byte vale zero e il secondo punta all'ultimo byte del file; se cioè il contenuto di questo puntatore è N, allora i byte da 2 a N contengono gli ultimi N-1 byte del file.

Letture e scrittura dei file

Se vogliamo scrivere un file sequenziale su disco, dobbiamo prima aprirlo con un comando OPEN, poi inviare i dati tramite istruzioni PRINT # e infine chiuderlo con un comando CLOSE. In tal caso la forma del comando OPEN è

OPEN f,8,c,"nome,SEQ,WRITE"
 dove f (numero file) può variare tra 1 e

127 e c (numero canale) deve essere compreso tra 2 e 14. SEQ e WRITE possono essere abbreviati con S e W. Questo comando viene inviato al DOS, che provvede a inserire nella Directory il nome e il tipo del file e cerca un blocco libero da indicare come primo blocco del file. Ora possiamo scrivere nel file usando l'istruzione

PRINT # f, dati

dove i dati possono essere numeri, stringhe tra apici, variabili ed espressioni di ogni tipo, esattamente come in un'istruzione PRINT sullo schermo: I dati vengono scritti dal DOS in uno dei buffer da 256 byte che si trovano all'interno del disk drive, e vengono scaricati su disco un blocco per volta. Quando abbiamo terminato dobbiamo chiudere il file con un comando

CLOSE f

Quando il DOS lo riceve riversa sul disco l'ultimo blocco del file, "chiude" il canale di comunicazione lasciando libero il buffer, alloca nella BAM i blocchi utilizzati e scrive nella Directory la lunghezza del file. È quindi fondamentale ricordarsi di chiudere i file aperti: in caso contrario l'informazione in essi contenuta può venir cancellata, dato che i blocchi registrati vengono ancora indicati come liberi nella BAM.

Se invece vogliamo leggere un file sequenziale da disco, dobbiamo prima aprirlo con un comando OPEN, poi ricevere i dati tramite istruzioni INPUT # e GET # e infine chiuderlo con un comando CLOSE. In questo caso la forma del comando OPEN è

OPEN f,8,c,"nome,SEQ,READ"

Per f e c valgono le limitazioni fatte precedentemente; SEQ e READ possono essere abbreviati con S e R. Questo comando viene inviato al DOS, che provvede a cercare nella Directory il file e a leggere l'indirizzo del primo blocco. Ora possiamo leggere dal file usando le istruzioni

INPUT # f, variabile
 GET # f, variabile

La prima istruzione assegna il successivo dato del file alla variabile indicata. I dati sono separati tra loro dai caratteri CHR\$(13) oppure CHR\$(44), cioè il RETURN o la virgola; si crea una condizione di errore, se si cerca di assegnare una stringa contenente caratteri non numerici a una variabile numerica, oppure tentando di leggere una stringa più lunga di 88 caratteri.

La seconda istruzione assegna il successivo byte del file alla variabile indicata; si crea una condizione di errore cercando di assegnare un carattere non numerico a una variabile numerica. Il DOS legge i dati dal file un blocco per volta e li pone in uno dei buffer da 256 byte che si trovano all'interno del disk drive. Quando abbiamo terminato dobbiamo chiudere il file con un comando

CLOSE f

Quando il DOS lo riceve "chiude" il canale di comunicazione, lasciando libero il buffer.

I file PRG vengono automaticamente aperti e chiusi in modo corretto dai comandi SAVE e LOAD.

Vediamo ora in dettaglio come vanno usate le istruzioni di scrittura (PRINT #) e di lettura (INPUT #, GET #). Innanzitutto diciamo che esse agiscono esattamente come le corrispondenti istruzioni PRINT/INPUT/GET, con la sola differenza che i dati vengono scritti nel file (e quindi nel nostro caso su disco) anziché sullo schermo, e vengono letti dal file (cioè dal disco) invece che dalla tastiera. Identico è l'effetto dei caratteri separatori, sul cui utilizzo ci soffermeremo brevemente per chiarire quale dev'essere la sintassi di un PRINT # e del corrispondente INPUT #, affinché i dati scritti vengano poi letti in modo esatto. Supponiamo di aver aperto, all'interno di un nostro programma, un file sequenziale chiamato "DATI" in scrittura, usando il comando

```
OPEN 8,8,8,"DATI,S,W"
```

e immaginiamo di aver assegnato le seguenti variabili:

```
R$=CHR$(13):V$=",":A$="PIPP0":  
B$="PROVA":A=53:B=12
```

Vogliamo ora scrivere nel file le variabili stringa A\$ e B\$ oppure le variabili numeriche A e B. Ci sono molti modi per farlo; ne esamineremo cinque, ai quali possono essere ricondotti in pratica tutti gli altri.

Primo modo - Si usano le seguenti istruzioni:

```
PRINT # 8,A$:PRINT # 8,B$
```

Analogamente all'istruzione PRINT, anche la PRINT # trasmette automaticamente un CHR\$(13) al termine del messaggio stampato, a meno che i dati da stampare siano seguiti da un punto e virgola (stampa di seguito) o da una virgola (tabulazione ogni 10 colonne). Il carattere CHR\$(13) è detto Carriage Return (ritorno carrello, cioè "vai a capo", termine che deriva dalle macchine da scrivere). Sullo schermo un CHR\$(13) fa andare a capo il cursore; in un file costituisce il principale carattere di separazione tra i "record", cioè i singoli dati. In base a quanto detto, eseguendo le istruzioni di cui sopra, nel file viene scritta la stringa "PIPP0", un CHR\$(13), la stringa "PROVA" e un CHR\$(13). In modo analogo per A e B usiamo le istruzioni

```
PRINT # 8,A$:PRINT # 8,B
```

Come in un'istruzione PRINT, i numeri vengono stampati preceduti da uno spazio (o dal segno meno) e seguiti ancora da uno spazio. Quindi in questo caso nel file viene scritta la stringa "b53b", un CHR\$(13), la stringa "b12b" e un CHR\$(13). La lettera "b" all'interno

delle stringhe rappresenta per chiarezza lo spazio.

Secondo modo - Si usa un'istruzione

```
PRINT # 8,A$;R$;B$
```

che ha esattamente lo stesso effetto delle istruzioni descritte nel punto precedente, in quanto anche qui le stringhe A\$ e B\$ vengono separate dal CHR\$(13) contenuto nella stringa R\$. Analogamente, l'istruzione

```
PRINT # 8,A;R$;B
```

stampa nel file le variabili A e B in modo identico a quello già descritto.

Terzo modo - Si usa un'istruzione

```
PRINT # 8,A$;V$;B$
```

L'unica differenza rispetto al modo precedente è che il carattere che separa A\$ da B\$ è la virgola contenuta nella stringa V\$, cioè CHR\$(44). Nel file viene quindi scritta la stringa "PIPP0", un CHR\$(44), la stringa "PROVA" e un CHR\$(13). Lo stesso discorso vale per

```
PRINT # 8,A;V$;B
```

in cui A e B risultano separate da un CHR\$(44) anziché da un CHR\$(13).

Quarto modo - Si usa un'istruzione

```
PRINT # 8,A$;B$
```

In questo caso non c'è nessun carattere di separazione, e nel file viene scritta la stringa "PIPP0", la stringa "PROVA" e un CHR\$(13). Analogamente, se usiamo

```
PRINT # 8,A;B
```

nel file viene scritta la stringa "b53b", la stringa "b12b" e un CHR\$(13).

Quinto modo - Si usa infine

un'istruzione

```
PRINT # 8,A$,B$
```

il cui risultato è quello di separare A\$ e B\$ con alcuni spazi, in quanto la virgola ha lo stesso effetto di tabulazione che avrebbe nell'istruzione PRINT. Nel file viene allora scritta la stringa "PIPP0", la stringa "bbbbbb", la stringa "PROVA" e un CHR\$(13). Analogamente, usando

```
PRINT # 8,A,B
```

scriviamo nel file la stringa "b53b", la stringa "bbbbbb", la stringa "b12b" e un CHR\$(13).

Ora che abbiamo elencato i cinque modi vediamo come possiamo leggere i dati che abbiamo scritto su disco. Supponiamo di aver aperto in lettura il file "DATI" con un comando

```
OPEN 8,8,8,"DATI,S,R"
```

e immaginiamo di voler leggere le variabili A\$ e B\$ scritte nei vari modi (vedremo che questo non sempre sarà possibile). Le istruzioni da usare sono

```
INPUT # 8,A$:INPUT # 8,B$
```

oppure

```
INPUT # 8,A$,B$
```

che hanno esattamente lo stesso effetto. Se A\$ e B\$ sono state scritte in uno qualsiasi dei primi tre modi, allora la lettura avviene con successo, assegnando ad A\$ la stringa "PIPP0" e a B\$ la stringa "PROVA". Infatti, sia CHR\$(13) che CHR\$(44) sono riconosciuti dall'istruzione INPUT # come caratteri "separatori" di record.

Se invece leggiamo i dati scritti nel quarto modo, ad A\$ viene assegnata la stringa "PIPP0PROVA" e a B\$ non viene assegnato nulla; ciò avviene perché i record non sono separati.

In modo simile, leggendo i dati scritti nel quinto modo, assegnamo ad A\$ la stringa "PIPP0bbbbbbPROVA" e B\$ resta vuota, in quanto gli spazi non sono riconosciuti come caratteri separatori. Un discorso analogo vale se leggiamo le variabili A e B con delle istruzioni

```
INPUT # 8,A:INPUT # 8,B
```

oppure

```
INPUT # 8,A,B
```

Se i dati erano stati scritti nei primi tre modi, allora ad A viene assegnato il valore 53 e a B il valore 12. Se invece erano stati utilizzati il quarto o il quinto modo, alla variabile A viene assegnato il valore 5312 e la variabile B è posta a zero, in quanto gli spazi vengono ignorati.

Abbiamo volutamente tralasciato l'istruzione GET # che, leggendo un solo carattere per volta, si presta ad applicazioni diverse da quelle viste finora. Ad esempio, consideriamo il breve programma

```
10 OPEN 8,8,8,"nome,S,R"  
20 GET#8,A$:IF ST=-128 THEN 40  
30 PRINT A$;:GOTO 20  
40 PRINT:CLOSE 8:END
```

Esso apre un file sequenziale e, byte per byte, legge da disco e stampa sullo schermo il contenuto del file. La variabile di stato delle operazioni di input/output, cioè ST, viene testata per verificare la fine del file (infatti ST normalmente vale zero, ma assume il valore -128 quando leggiamo l'ultimo carattere del file). Un programmino di questo tipo può servire ad "esplorare" un file di cui non conosciamo il contenuto.

Vediamo infine un altro uso dell'istruzione GET #. Supponiamo di voler salvare su disco una serie di N numeri, tutti compresi tra zero e 255 (potrebbero essere ad esempio codici di schermo o codici ASCII). Costruiamo un programma con tali numeri memorizzati in opportune istruzioni DATA, e aggiungiamo le seguenti linee:

```
10 OPEN 8,8,8,"CODICI,S,W"  
20 FOR I=1 TO N:READ A:PRINT#8,CHR$(A);:NEXT  
30 CLOSE 8
```

In tal modo costruiamo un file "CODI-

CI", in cui ogni numero occupa un solo byte, essendo scritto nella forma CHR\$(n). Inoltre non sono necessari caratteri separatori (infatti la PRINT # termina con un punto e virgola), in quanto il file verrà poi letto carattere per carattere.

Se la linea 20 fosse stata

```
20 FOR I=1 TO N:READ A: PRINT #
8,A:NEXT
```

cioè se avessimo scritto i numeri "per esteso", in modo da poterli leggere con delle istruzioni INPUT #, il file "CODICI" sarebbe stato almeno cinque volte più lungo, con corrispondente aumento dello spazio occupato sul disco e del tempo di lettura/scrittura.

Supponiamo ora di voler leggere il file, assegnando i numeri a un vettore V%(i). La procedura da utilizzare è la seguente:

```
10 DIM V%(N):OPEN 8,8,8,"CODICI,S,R"
20 FOR I=1 TO N:GET#8,A$:V%(I)=ASC(A$):NEXT
30 CLOSE 8
```

Il programma copia-file

In aggiunta a questo articolo forniamo un breve programma, nelle due versioni per C16-Plus/4 e VIC 20, che consente di trasferire da un disco all'altro file di qualsiasi tipo (tranne i relativi). Questo copia-file crea nella RAM del computer, modificando i puntatori del BASIC, un buffer, cioè un'area di memoria che viene utilizzata per "immagazzinare" temporaneamente i file letti dal disco sorgente, in modo da poterli poi riversare sul disco destinazione. La capacità di questo buffer limita la lunghezza massima dei file trasferibili, nel senso che non è possibile copiare un file più lungo del buffer. Premesso che per utilizzare questo programma con il VIC 20 è necessario possedere un'espansione di memoria da almeno 3 Kilobyte, diamo qui di seguito le capacità del buffer per i vari computer e nelle varie configurazioni:

C1630 blocchi
Plus/495 blocchi
VIC 20 + 3K ...10 blocchi
VIC 20 + 8K ...30 blocchi
VIC 20 + 16K ..62 blocchi

È possibile trasferire fino a 30 file per volta e se la lunghezza complessiva dei file da copiare supera la capacità del buffer, il trasferimento avviene in più passate, leggendo ogni volta un numero di file tale da poter essere contenuto in memoria.

Il programma è gestito tramite un semplice menu che comprende le seguenti opzioni:

DIRECTORY, che permette di visualizzare sullo schermo, divisa in pagine, la Directory del disco contenuto nel drive.

COMANDO DISCO, per inviare qualsiasi comando al drive (seguendo la normale sintassi, ad esempio I per Initialize, V per Validate, ecc.).

COPIA FILE, che consente di selezionare, visualizzando ad uno ad uno i nomi dei file contenuti sul disco, se un certo file deve essere copiato (in tal caso premere S) o no (premere N); se si tenta di scegliere un file di dimensioni supe-

riori al buffer, viene visualizzato il messaggio "Troppo lungo!". Se si desidera annullare la selezione perché si è commesso un errore, premere la barra spaziatrice. Il programma provvede a segnalare quando va inserito nel drive il disco sorgente e quando va invece inserito il disco destinazione.

FINE, da usare al termine delle operazioni. È importante uscire dal programma in questo modo e non usando il tasto RUN/STOP, in quanto questa opzione provvede a ristabilire i puntatori del BASIC prima di terminare il programma.

Lo status del drive viene mostrato al termine di qualsiasi operazione eseguita sul disco, ed è sempre visibile nella parte bassa dello schermo contenente il menu. Se si verifica una condizione di errore durante il trasferimento dei file, l'operazione viene troncata.

La lettura e la scrittura dei file avviene byte per byte, con l'utilizzazione di due sottoprogrammi in linguaggio macchina, che richiamano opportunamente le routine CHRIN e CHROUT del sistema operativo Kernal.

Per concludere illustriamo brevemente la divisione in blocchi del programma:

Linee	Commento
10- 30	Inizializzazione - Modifica puntatori BASIC
40-110	Stampa menu - Input comandi
120-360	Lettura Directory - Selezione file
370-480	Lettura file selezionati
490-610	Scrittura file selezionati
620-650	Attesa tasto premuto
660-720	Input "S" oppure "N" per selezione file
730-750	Lettura status disco
760-830	Input e invio comando disco
840-870	Uscita dal programma - Ripristino puntatori BASIC
880-910	Stampa messaggi cambio disco - Inizializzazione disco
920-960	Dati linguaggio macchina

Copia File Versione per VIC20

```
10 POKE950,PEEK(56):POKE55,0:POKE56,PEEK(
44)+16:CLR:EM=PEEK(56):EE=PEEK(950)
:rem 244
15 TB=(EE*256-(EM*256))/256 :rem 126
20 DIML%(31,2),C$(31,1):OPEN8,8,15:Q$="
{ 8 GIU' }" :rem 183
30 FORI=848TO917:READA:POKEI,A:NEXT:POKE3
6879,25 :rem 244
40 FL=0:PRINT"{CLR}{CYN}{GIU'}{RVS} C O P
I A{ 4 SPAZI}F I L E " :rem 79
50 PRINT"{ 2 GIU'}{BLK}{ 2 SPAZI}{RVS} 1
{OFF}{ 2 SPAZI}DIRECTORY" :rem 114
60 PRINT"{ 2 GIU'}{ 2 SPAZI}{RVS} 2 {OFF}
{ 2 SPAZI}COMANDO DISCO" :rem 162
70 PRINT"{ 2 GIU'}{ 2 SPAZI}{RVS} 3 {OFF}
{ 2 SPAZI}COPIA FILE" :rem 189
```

```
80 PRINT"{ 2 GIU'}{ 2 SPAZI}{RVS} 4 {OFF}
{ 2 SPAZI}FINE{ 6 GIU'}" :rem 187
90 GOSUB730:GOSUB740 :rem 216
100 GOSUB640:ONVAL(C$)GOTO130,760,120,840
:rem 58
110 GOTO90 :rem 52
120 FL=1:GOSUB880:IFVAL(E$)>0THENGOSUB740
:GOSUB640:GOTO40 :rem 54
130 OPEN1,8,0,"$0" :rem 123
140 GOSUB730:IFVAL(E$)>0THENGOSUB740:CLOS
E1:GOTO100 :rem 178
150 P=0:Q=0:PRINT"{CLR}{BLK}"; :rem 180
160 GET#1,A$,B$ :rem 238
170 GET#1,A$,B$ :rem 239
180 GET#1,A$,B$ :rem 240
190 C=0 :rem 74
200 IFA$<>" "THENC=ASC(A$) :rem 128
210 IFB$<>" "THENC=C+ASC(B$)*256 :rem 184
220 P$="":P=P+1:PRINT"{RVS}"MID$(STR$(C),
```

```

2) ;TAB(4) ;:IFP>1 THENPRINT "{OFF}";
:rem 158
230 GET#1, B$:IFPEEK(144)<>0 THEN350
:rem 164
240 IFB$<>CHR$(34) THEN230 :rem 130
250 GET#1, B$:IFB$<>CHR$(34) THENPRINTB$;:P
$=P$+B$:GOTO250 :rem 22
260 GET#1, B$:IFB$=CHR$(32) THEN260
:rem 72
270 PRINT:K$="" :rem 80
280 K$=K$+B$:GET#1, B$:IFB$<>"" THEN280
:rem 252
290 PRINT "{RVS}{ 4 DES}"LEFT$(K$,3):IFFL=
1ANDP>1 THENGOSUB660:IFC$="" THENCLOSE
1:GOTO40 :rem 100
300 IFFL=1ANDQ=30 THEN360 :rem 174
310 IFP/10<>INT(P/10)ORFL=1 THEN340
:rem 0
320 IFFL=0 THENGOSUB620 :rem 106

330 PRINT "{CLR}"; :rem 53
340 IFPEEK(144)=0 THEN170 :rem 105
350 PRINT "{RVS}BLOCKS FREE" :rem 91
360 CLOSE1:IFFL=0 THENGOSUB620:GOTO40
:rem 38
370 IFQ=0 THEN40 :rem 122
380 I=1 :rem 82
390 PRINT "{CLR} [<6>]":NB=0:II=I:POKE883,0
:POKE884,EM:POKE857,0:POKE858,EM
:rem 138
400 L%(I,1)=PEEK(857):L%(I,2)=PEEK(858):N
B=NB+L%(I,0):IFNB>TBORI>Q THEN490
:rem 233
410 PRINT"LETT.{RVS}"C$(I,0) :rem 60
420 OPEN4,8,4,C$(I,0)+",""+C$(I,1)+","R"
:rem 90
430 GOSUB730:IFVAL(E$)=0 THEN450 :rem 76
440 CLOSE4:PRINT:GOSUB750:PRINT"
{ 2 GIU' }{BLU}{RVS}PREMI UN TASTO":GO
SUB640:GOTO40 :rem 147
450 SYS848:SYS65511:OPEN8,8,15 :rem 17
460 GOSUB730:IFVAL(E$)=0 THEN480 :rem 82
470 PRINT:GOSUB750:PRINT "{ 2 GIU' }{BLU}
{RVS}PREMI UN TASTO":GOSUB640:GOTO40
:rem 178
480 I=I+1:GOTO400 :rem 206
490 GOSUB900:IFVAL(E$)>0 THENPRINT:GOSUB75
0:GOSUB640:GOTO40 :rem 199
500 PRINT "{CLR} [<6>]":J=II :rem 229
510 PRINT"SCR.{RVS}"C$(J,0):POKE840,L%(J+
1,1):POKE841,L%(J+1,2) :rem 219
520 OPEN4,8,4,C$(J,0)+",""+C$(J,1)+","W"
:rem 98
530 GOSUB730:IFVAL(E$)=0 THEN550 :rem 78
540 CLOSE4:PRINT:GOSUB750:PRINT"
{ 2 GIU' }{BLU}{RVS}PREMI UN TASTO":GO
SUB640:GOTO40 :rem 148
550 SYS877:SYS65511:OPEN8,8,15 :rem 20
560 GOSUB730:IFVAL(E$)=0 THEN580 :rem 84
570 PRINT:GOSUB750:PRINT "{ 2 GIU' }{BLU}
{RVS}PREMI UN TASTO":GOSUB640:GOTO40
:rem 179
580 J=J+1:IFJ<I THEN510 :rem 39
590 IFJ<Q THENGOSUB880:IFVAL(E$)>0 THENPRIN
T:GOSUB750:GOSUB640:GOTO40 :rem 100
600 IFJ<Q THEN390 :rem 199
610 GOTO40 :rem 52
620 PRINT "{HOME}"; :rem 183
630 PRINTQ$Q$"{BLU}{ 5 GIU' }{ 4 DES }{RVS}
PREMI UN TASTO{BLK}" :rem 137

```

```

640 GETC$:IFC$="" THEN640 :rem 91
650 RETURN :rem 123
660 POKE198,0 :rem 201
670 GETC$:IFC$<>"S"ANDC$<>"N"ANDC$<>" "TH
EN670 :rem 47
680 IFC$="" THENRETURN :rem 36
690 IFC$="N" THENPRINTTAB(8) "{SU}{PUR}
{RVS}NO{BLK}":RETURN :rem 74
700 IFC>TB THENPRINT "{GIU' }{RED}{RVS}
{ 4 DES}TROPPO LUNGO!{GIU' }{BLK}":RET
URN :rem 53
710 PRINTTAB(8) "{SU}{GRN}{RVS}SI{BLK}":Q=
Q+1:C$(Q,0)=P$:C$(Q,1)=LEFT$(K$,1)
:rem 196
720 L%(Q,0)=C:RETURN :rem 162
730 INPUT#8,E$,L$,M$,N$:RETURN :rem 11
740 PRINT "{HOME}{ 20 GIU' }{ 43 SPAZI}
{ 4 SU}" :rem 23
750 PRINT "{RED}STATUS:":PRINT "{RVS}{GIU' }
"E$"{DES}"L$"{DES}"M$"{DES}"N$:RETURN
:rem 206
760 CC$="":PRINT "{CLR} [<6>] {GRN}"Q$"COMAN
DO?{GIU' }" :rem 39
770 GOSUB640:IFC$<>CHR$(13)ANDC$<>CHR$(20
) THENCC$=CC$+C$ :rem 210
780 IFCC$>"" THENPRINTC$; :rem 184
790 IFC$=CHR$(20)ANDCC$>"" THENCC$=LEFT$(C
C$,LEN(CC$)-1) :rem 74
800 IFC$<>CHR$(13)ANDLEN(CC$)<44 THEN770
:rem 220
810 IFLEN(CC$)=21 THENPRINT :rem 94
820 PRINT#8,CC$:PRINT "{GIU' }":GOSUB730:GO
SUB750 :rem 30
830 GOSUB620:GOTO40 :rem 138
840 PRINT "{CLR}{BLU}{ 2 GIU' }SEI SICURO (
S/N)?" :rem 87
850 GETC$:IFC$<>"S"ANDC$<>"N" THEN850
:rem 55
860 IFC$="N" THEN40 :rem 246
870 POKE56,PEEK(950):CLR:PRINT "{CLR}":END
:rem 67
880 PRINT "{CLR}{GRN}"Q$"{ 2 GIU' }{RVS}INS
ERISCI{ 2 SPAZI}DISCO SORG.":rem 147
890 PRINT "{HOME}";:GOSUB630:PRINT#8,"I0":
GOSUB730:RETURN :rem 140
900 PRINT "{CLR}{GRN}"Q$"{ 2 GIU' }{RVS}INS
ERISCI{ 2 SPAZI}DISCO DEST.":rem 129
910 PRINT "{HOME}";:GOSUB630:PRINT#8,"I0":
GOSUB730:RETURN :rem 133
920 DATA162,4,32,198,255,32,207,255,141,0
,24,238,89,3 :rem 104
930 DATA208,3,238,90,3,165,144,240,238,16
9,4,32,195,255 :rem 212
940 DATA96,162,4,32,201,255,173,0,24,32,2
10,255,238,115 :rem 188
950 DATA3,208,3,238,116,3,173,72,3,205,11
5,3,208,234 :rem 39
960 DATA173,73,3,205,116,3,208,226,169,4,
32,195,255,96 :rem 166

```

Copia File Versione per C16-Plus/4

```

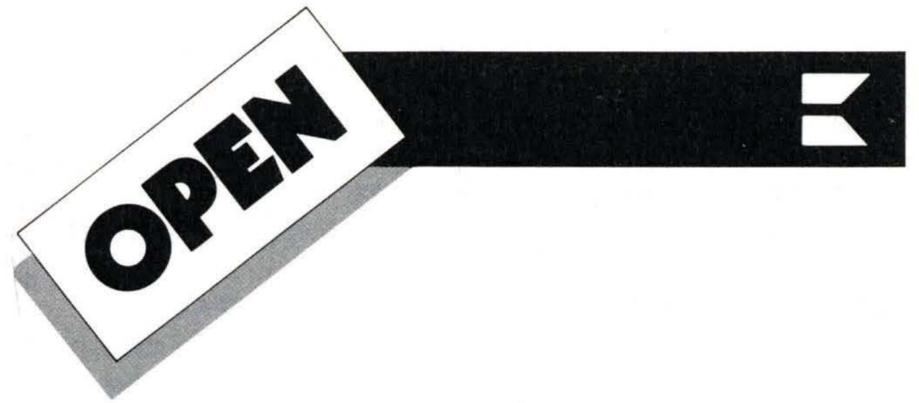
10 POKE950,PEEK(56):POKE55,0:POKE56,33:CL
R:EM=PEEK(56):EE=PEEK(950) :rem 234
15 IFEE>128 THENEE=128 :rem 233
20 TB=(EE*256-(EM*256))/256 :rem 122
25 DIML%(31,2),C$(31,1):OPEN8,8,15:Q$="
{ 8 GIU' }" :rem 188

```

```

30 FORI=848TO917:READA:POKEI,A:NEXT:COLOR
   0,1:COLOR4,2,1 :rem 213
40 FL=0:PRINT"{CLR}{CYN}{ 3 GIU' }"TAB(8)"
   {RVS}{ 2 SPAZI}C O P I A{ 4 SPAZI}F I
   L E{ 2 SPAZI}" :rem 21
50 PRINT"{ 3 GIU' }{BLU}"TAB(11)"{RVS} 1
   {OFF}{ 2 SPAZI}DIRECTORY" :rem 224
60 PRINT"{ 2 GIU' }"TAB(11)"{RVS} 2 {OFF}
   { 2 SPAZI}COMANDO DISCO" :rem 112
70 PRINT"{ 2 GIU' }"TAB(11)"{RVS} 3 {OFF}
   { 2 SPAZI}COPIA FILE" :rem 139
80 PRINT"{ 2 GIU' }"TAB(11)"{RVS} 4 {OFF}
   { 2 SPAZI}FINE{ 5 GIU' }" :rem 120
90 GOSUB730:GOSUB740 :rem 216
100 GOSUB640:ONVAL(C$)GOTO130,760,120,840
   :rem 58
110 GOTO90 :rem 52
120 FL=1:GOSUB880:IFVAL(E$)>0THENPRINT"
   { 4 SU}":GOSUB740:GOSUB640:GOTO40
   :rem 133
130 OPEN1,8,0,"$0" :rem 123
140 GOSUB730:IFVAL(E$)>0THENGOSUB740:CLOS
   E1:GOTO100 :rem 178
150 P=0:Q=0:PRINT"{CLR}{BLU}" :rem 8
160 GET#1,A$,B$ :rem 238
170 GET#1,A$,B$ :rem 239
180 GET#1,A$,B$ :rem 240
190 C=0 :rem 74
200 IFA$<>" "THENC=ASC(A$) :rem 128
210 IFB$<>" "THENC=C+ASC(B$)*256 :rem 184
220 P$="":P=P+1:PRINT"{RVS}"MID$(STR$(C),
   2);TAB(4);:IFP>1THENPRINT"{OFF}";
   :rem 158
230 GET#1,B$:IFPEEK(144)<>0THEN350
   :rem 164
240 IFB$<>CHR$(34)THEN230 :rem 130
250 GET#1,B$:IFB$<>CHR$(34)THENPRINTB$;:P
   $=P$+B$:GOTO250 :rem 22
260 GET#1,B$:IFB$=CHR$(32)THEN260
   :rem 72
270 PRINTTAB(21);:K$="" :rem 22
280 K$=K$+B$:GET#1,B$:IFB$<>" "THEN280
   :rem 252
290 PRINT"{RVS}"LEFT$(K$,3):IFFL=1ANDP>1T
   HENGOSUB660:IFC$=" "THENCLOSE1:GOTO40
   :rem 240
300 IFFL=1ANDQ=30THEN360 :rem 174
310 IFP/21<>INT(P/21)ORFL=1THEN340
   :rem 4
320 IFFL=0THENGOSUB620 :rem 106
330 PRINT"{CLR}" :rem 250
340 IFPEEK(144)=0THEN170 :rem 105
350 PRINT"{RVS}BLOCKS FREE" :rem 91
360 CLOSE1:IFFL=0THENGOSUB620:GOTO40
   :rem 38
370 IFQ=0THEN40 :rem 122
380 I=1 :rem 82
390 PRINT"{CLR} [<6>]":NB=0:II=I:POKE883,0
   :POKE884,EM:POKE857,0:POKE858,EM
   :rem 138
400 L$(I,1)=PEEK(857):L$(I,2)=PEEK(858):N
   B=NB+L$(I,0):IFNB>TBORI>QTHEN490
   :rem 233
410 PRINT"STO LEGGENDO{ 2 SPAZI}{RVS}"C$(
   I,0) :rem 16
420 OPEN4,8,4,C$(I,0)+", "+C$(I,1)+",R"
   :rem 90
430 GOSUB730:IFVAL(E$)=0THEN450 :rem 76
440 CLOSE4:PRINT"{ 2 GIU' }":GOSUB740:PRIN
   T"{ 2 GIU' } [<1>]{RVS}PREMI UN TASTO":

```



```

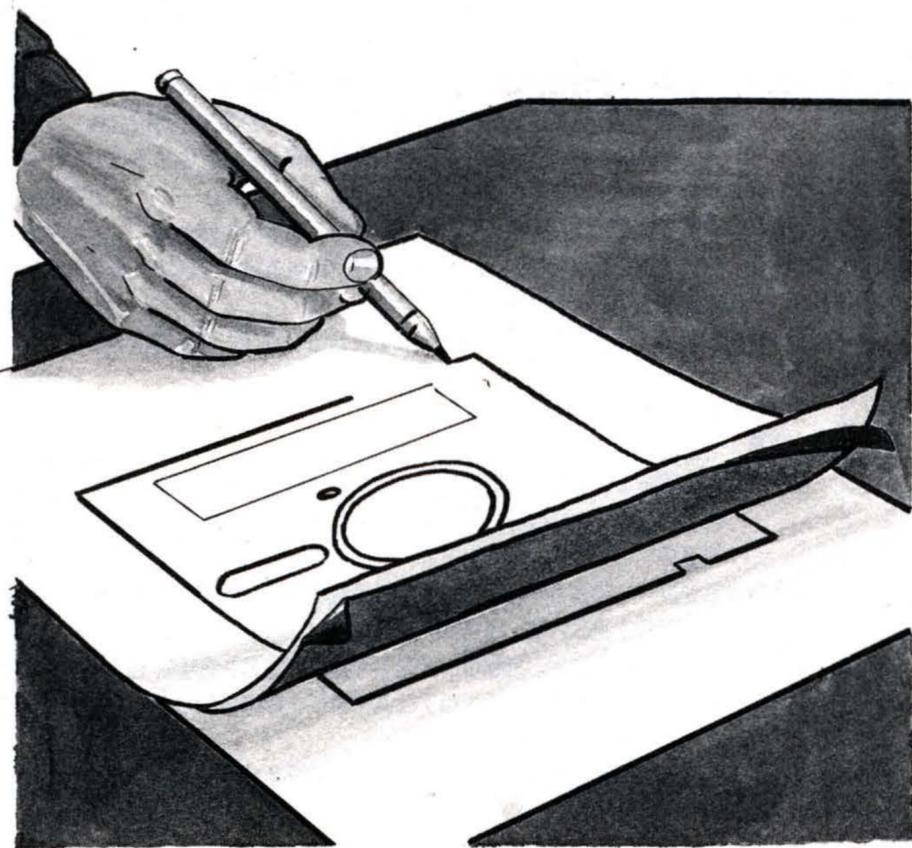
GOSUB640:GOTO40 :rem 90
450 SYS848:SYS65511:OPEN8,8,15 :rem 17
460 GOSUB730:IFVAL(E$)=0THEN480 :rem 82
470 PRINT"{ 2 GIU' }":GOSUB740:PRINT"
   { 2 GIU' } [<1>]{RVS}PREMI UN TASTO":GO
   SUB640:GOTO40 :rem 121
480 I=I+1:GOTO400 :rem 206
490 GOSUB900:IFVAL(E$)>0THENPRINT"
   { 4 SU}":GOSUB740:GOSUB640:GOTO40
   :rem 78
500 PRINT"{CLR} [<6>]":J=II :rem 229
510 PRINT"STO SCRIVENDO{ 2 SPAZI}{RVS}"C$(
   J,0):POKE840,L$(J+1,1):POKE841,L$(J+
   1,2) :rem 104
520 OPEN4,8,4,C$(J,0)+", "+C$(J,1)+",W"
   :rem 98
530 GOSUB730:IFVAL(E$)=0THEN550 :rem 78
540 CLOSE4:PRINT"{ 2 GIU' }":GOSUB740:PRIN
   T"{ 2 GIU' } [<1>]{RVS}PREMI UN TASTO":
   GOSUB640:GOTO40 :rem 91
550 SYS877:SYS65511:OPEN8,8,15 :rem 20
560 GOSUB730:IFVAL(E$)=0THEN580 :rem 84
570 PRINT"{ 2 GIU' }":GOSUB740:PRINT"
   { 2 GIU' } [<1>]{RVS}PREMI UN TASTO":GO
   SUB640:GOTO40 :rem 122
580 J=J+1:IFJ<I THEN510 :rem 39
590 IFJ<QTHENGOSUB880:IFVAL(E$)>0THENPRIN
   T"{ 4 SU}":GOSUB740:GOSUB640:GOTO40
   :rem 235
600 IFJ<QTHEN390 :rem 199
610 GOTO40 :rem 52
620 PRINT"{HOME}"; :rem 183
630 PRINTQ$Q$" [<1>]{ 7 GIU' }{ 4 DES}{RVS}
   PREMI UN TASTO{BLU}" :rem 156
640 GETKEYC$ :rem 202
650 RETURN :rem 123
660 POKE239,0 :rem 197
670 GETC$:IFC$<>"S"ANDC$<>"N"ANDC$<>" "TH
   EN670 :rem 47
680 IFC$=" "THENRETURN :rem 36
690 IFC$="N"THENPRINTTAB(25)"{SU}{PUR}
   {RVS}NO{BLU}":RETURN :rem 8
700 IFC>TBTHENPRINT"{GIU' }{RED}{RVS}
   { 4 DES}TROPPO LUNGO!{GIU' }{BLU}":RET
   URN :rem 196
710 PRINTTAB(25)"{SU}{GRN}{RVS}SI{BLU}":Q
   =Q+1:C$(Q,0)=P$:C$(Q,1)=LEFT$(K$,1)
   :rem 130
720 L$(Q,0)=C:RETURN :rem 162
730 INPUT#8,E$,L$,M$,N$:RETURN :rem 11
740 PRINT"{SU}{ 39 SPAZI}{ 3 SU}"
   :rem 176
750 PRINT"{ 2 GIU' }{YEL}STATUS:{RVS}"E$"
   {DES}"L$"{DES}"M$"{DES}"N$:RETURN
   :rem 86

```

```

760 CC$="":PRINT"{CLR}[<6>]"Q$"COMANDO?
      {GIU' }"           :rem 9
770 GOSUB640:IFC$<>CHR$(13)ANDC$<>CHR$(20
      )THENCC$=CC$+C$   :rem 210
780 IFCC$>""THENPRINTC$; :rem 184
790 IFC$=CHR$(20)ANDCC$>""THENCC$=LEFT$(C
      C$,LEN(CC$)-1)    :rem 74
800 IFC$<>CHR$(13)ANDLEN(CC$)<39THEN770
      :rem 224
810 IFLEN(CC$)=39THENPRINT :rem 103
820 PRINT#8,CC$:PRINT"{ 3 GIU' }":GOSUB730
      :GOSUB740         :rem 63
830 GOSUB620:GOTO40     :rem 138
840 PRINT"{CLR}{YEL}{ 2 GIU' }SEI SICURO (
      S/N)?"           :rem 214
850 GETC$:IFC$<>"S"ANDC$<>"N"THEN850
      :rem 55
860 IFC$="N"THEN40     :rem 246
870 POKE56,PEEK(950):CLR:SCNCLR:END
      :rem 164
880 PRINT"{CLR}{YEL}"Q$"{ 4 GIU' }
      { 8 DES}{RVS}INSERISCI DISCO SORGENTE
      "                 :rem 27
890 PRINT"{HOME}"TAB(9);:GOSUB630:PRINT#8
      ,"I0":GOSUB730:RETURN :rem 237
900 PRINT"{CLR}{YEL}"Q$"{ 4 GIU' }
      { 6 DES}{RVS}INSERISCI DISCO DESTINAZ
      IONE"             :rem 0
910 PRINT"{HOME}"TAB(9);:GOSUB630:PRINT#8
      ,"I0":GOSUB730:RETURN :rem 230
920 DATA162,4,32,198,255,32,207,255,141,0
      ,24,238,89,3     :rem 104
930 DATA208,3,238,90,3,165,144,240,238,16
      9,4,32,195,255   :rem 212
940 DATA96,162,4,32,201,255,173,0,24,32,2
      10,255,238,115   :rem 188
950 DATA3,208,3,238,116,3,173,72,3,205,11
      5,3,208,234     :rem 39
960 DATA173,73,3,205,116,3,208,226,169,4,
      32,195,255,96   :rem 166

```



SUPERCOMMODORE

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA da compilare e spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano - Tel. 6880951/2/3/4/5

Vogliate inviarmi i numeri
 di SUPERCOMMODORE con cassettaAnno
 Al prezzo di L. 15.000 cad.

- Contanti allegati Assegno allegato n°
- Ho spedito l'importo a mezzo vaglia postale
- Ho versato l'importo sul ccp. n° 11666203 intestato Gruppo Editoriale Jackson - Milano
- Pagherò in contrassegno al postino al ricevimento

BUONO D'ORDINE PER RIVISTE ARRETRATE
 Anche se l'ordine riguardasse la sola cassetta, questa verrà comunque inviata insieme alla rivista al prezzo indicato.

Nome
 Cognome
 Via
 CittàC.A.P.Prov.
 Se richiesta fattura:
 Cod. F. e P. Iva
 Data
 Firma
 Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

MLX per VIC 20

di C. Brannon
trad. e adatt.
di F. Stella

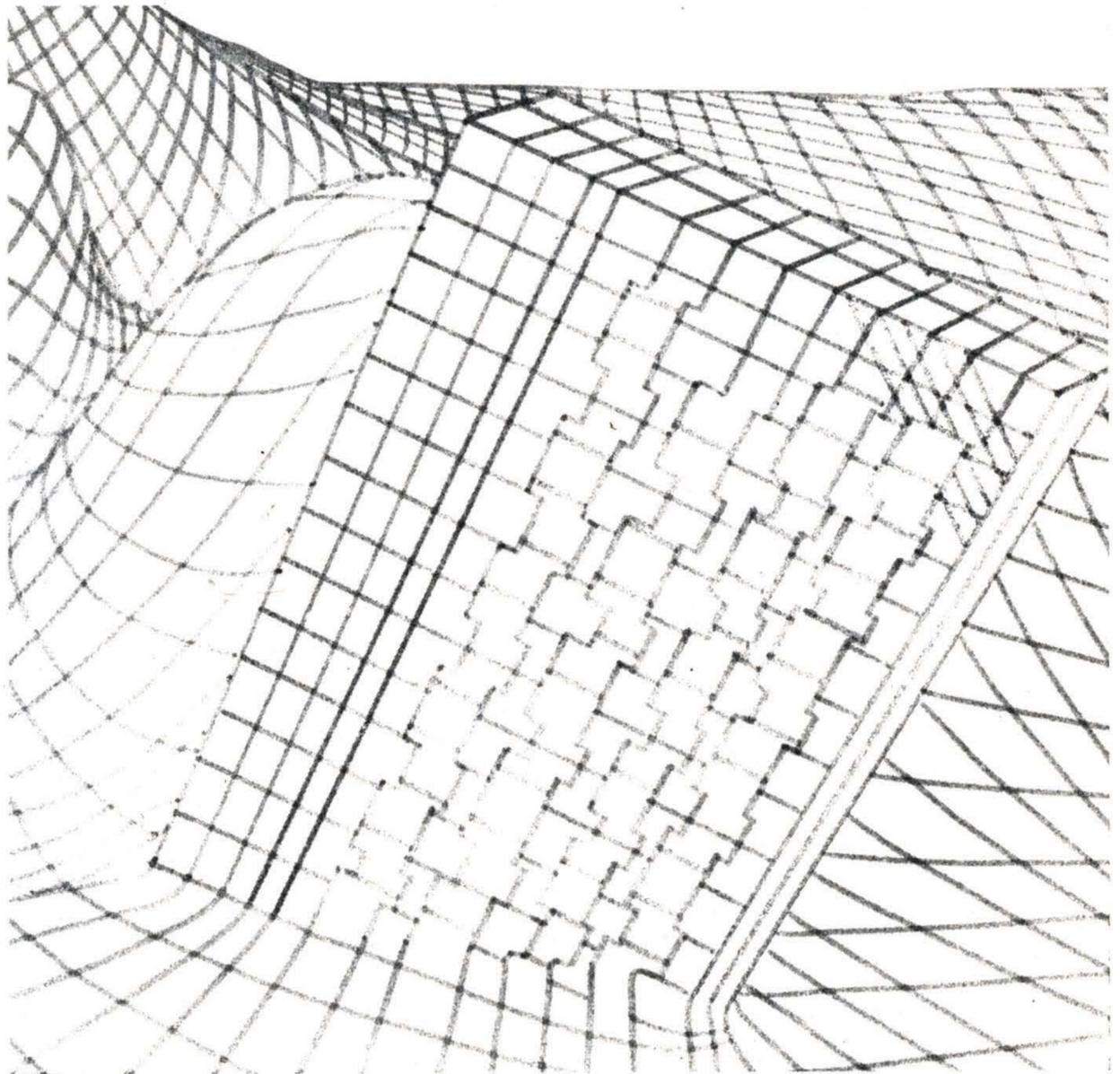
MLX è un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e quello della locazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato. MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali:

SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed è utile nel caso si inseriscano i listati in più parti.



SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione

LOAD "nome del file",1,1 per il registratore, oppure LOAD "nome del file",8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

MLX versione per VIC 20

```
100 PRINT" {CLR} {PUR}";CHR$(142);CHR$(8);
      :rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
      :rem 144
110 PRINT" {RVS} { 14 SPAZI}"
      :rem 117
120 PRINT" {RVS} {DES} {OFF} {<*>} {RVS}
      {DES} {DES} { 2 SPAZI} {<*>} {OFF} {<*>}
      {RVS} {RVS} "
      :rem 191
130 PRINT" {RVS} {DES} {<G>} {DES}
      { 2 DES} {OFF} {RVS} {<*>} {OFF} {<*>}
      {RVS} {OFF}"
      :rem 122
140 PRINT" {RVS} { 14 SPAZI}"
      :rem 120
200 PRINT" { 2 GIU' } {PUR} {BLK} UN PROGRAMMA
      PER L'IN-TRODUZIONE { 2 SPAZI} DI ROUT
      INEIN LINGUAGGIO MACCHINA"; :rem 202
205 PRINT" A PROVA DI ERRORE { 5 GIU' }"
      :rem 226
210 PRINT" {BLK} { 4 SU} INDIRIZZO DI PARTEN
      ZA": INPUTS:F=1-F:C$=CHR$(31+119*F)
      :rem 52
220 IFS<256ORS>32767THENGOSUB3000:GOTO210
      :rem 2
225 PRINT:PRINT:PRINT:PRINT
      :rem 123
230 PRINT" {BLK} { 3 SU} INDIRIZZO CONCLUSIV
      O": INPUTE:F=1-F:C$=CHR$(31+119*F)
      :rem 170
```

```
240 IFE<256ORE>32767THENGOSUB3000:GOTO230
      :rem 234
250 IFE<STHENPRINTC$;" {RVS} INDIRIZZO CONC
      LUSIVO <INDIRIZZO DI PARTENZA
      { 3 GIU' }":GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT
      :rem 179
300 PRINT" {CLR}";CHR$(14):AD=S
      :rem 56
310 PRINTRIGHT$ ("0000"+MID$(STR$(AD),2),5
      );":":FORJ=1TO6
      :rem 234
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
      :rem 228
390 IFN=-211THEN710
      :rem 62
400 IFN=-204THEN790
      :rem 64
410 IFN=-206THENPRINT:PRINT" {GIU' } INSERIS
      CI UN NUOVO { 4 SPAZI} INDIRIZZO":INPU
      TZZ
      :rem 42
415 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
      {RVS} ESCE DAL CAMPO DI VA-" :rem 150
416 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
      {RVS} LORI INDICATO":GOSUB1000:GOTO410
      :rem 105
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
      :rem 238
420 IFN<>-196THEN480
      :rem 133
430 PRINT:INPUT" LISTATO:DA";F:PRINT,"
      { 2 SIN}A":INPUTT
      :rem 29
```

```

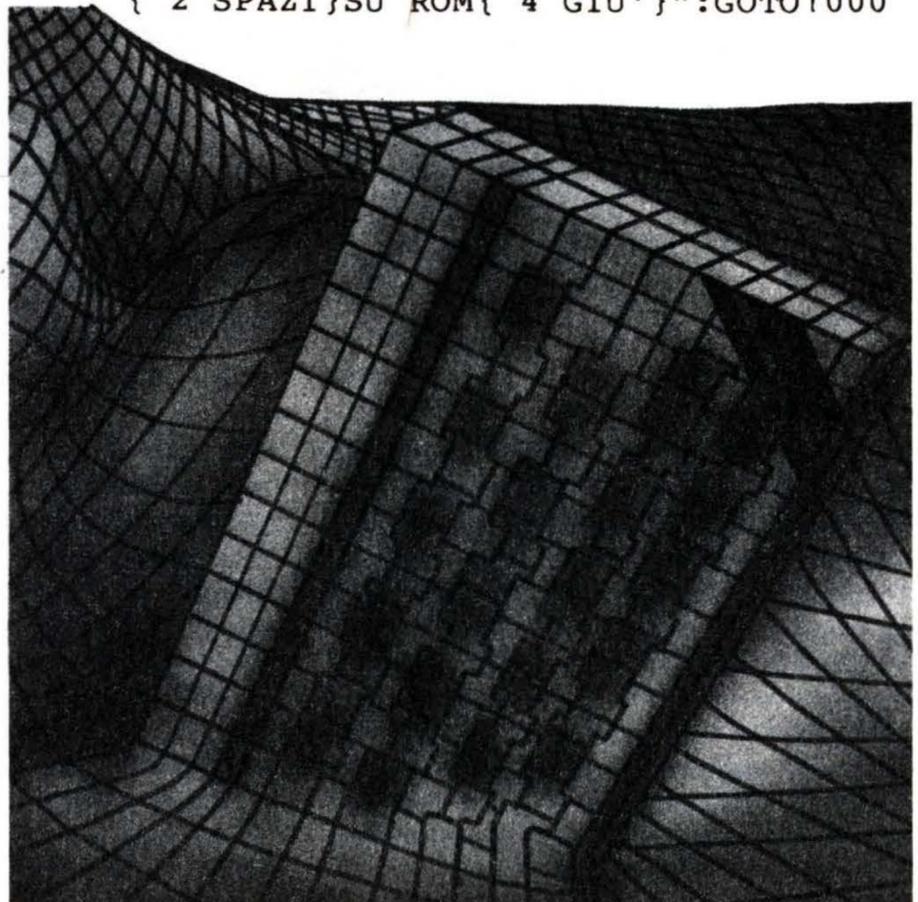
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
;S;"{SIN}, MASSIMO";E:GOTO430
:rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
0"+MID$(STR$(I),2),5);":":rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3THENPRINTS
PC(10);:rem 34
457 PRINTRIGHT$("00"+MID$(STR$(N),2),3);"
,";:rem 157
460 GETA$:IFA$>" THENPRINT:PRINT:GOTO310
:rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310:rem 50
480 IFN<0THENPRINT:GOTO310:rem 168
490 A(J)=N:NEXTJ:rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT:rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
:rem 234
515 IFN=CKSUMTHEN530:rem 255
520 PRINT:PRINT"LA RIGA E' STATA INSE-
RITA IN MANIERA ERRATA":rem 39
525 PRINT"RI-INSEIRLA":PRINT:GOSUB1000:G
OTO310:rem 167
530 GOSUB2000:rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
:rem 80
550 AD=AD+6:IFAD<ETHEN310:rem 212
560 GOTO710:rem 108
570 N=0:Z=0:rem 88
580 PRINT" [<+>]";:rem 79
581 GETA$:IFA$="" THEN581:rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
ORA=32THEN670:rem 229
590 IFA>128THENN=-A:RETURN:rem 137
600 IFA<>20THEN630:rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
{SIN} {SIN}";:GOTO690:rem 172
620 GOTO570:rem 109
630 IFA<48ORA>57THEN580:rem 105
640 PRINTA$;:N=N*10+A-48:rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
:rem 229
660 Z=Z+1:IFZ<3THEN580:rem 71
670 IFZ=0THENGOSUB1000:GOTO570:rem 114
680 PRINT",":RETURN:rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
:rem 149
692 FORI=1TO3:T=PEEK(S%-I):rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
:rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
:rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
{ 3 GIU}'":rem 236
720 INPUT"{GIU}' NOME DEL FILE";F$:
:rem 199
730 PRINT:PRINT"{ 2 GIU}'{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)":rem 128
740 GETA$:IFA$<>"N"ANDA$<>"D" THEN740
:rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
:rem 158
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256:rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469:rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
66:rem 69
765 POKE254,S/256:POKE253,S-PEEK(254)*256
:POKE780,253:rem 12

```

```

766 POKE782,E/256:POKE781,E-PEEK(782)*256
:SYS65496:rem 124
770 IF(PEEK(783)AND1)OR(ST AND191) THEN780
:rem 111
775 PRINT"{GIU}' OPERAZIONE CONCLUSA.":END
:rem 152
780 PRINT"{GIU}'ERRORE DI REGISTRAZIO-NE.
RIPROVA.":IFDV=1 THEN720:rem 8
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
;E2$:CLOSE15:GOTO720:rem 103
782 GOTO720:rem 115
790 PRINT"{CLR}{RVS}*** LOAD ***
{ 2 GIU}'":rem 212
800 INPUT"{ 2 GIU}' NOME DEL FILE";F$:
:rem 215
810 PRINT:PRINT"{ 2 GIU}'{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)":rem 127
820 GETA$:IFA$<>"N"ANDA$<>"D" THEN820
:rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
:rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256:rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469:rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
66:rem 70
850 POKE780,0:SYS65493:rem 11
860 IF(PEEK(783)AND1)OR(ST AND191) THEN870
:rem 111
865 PRINT"{GIU}' OPERAZIONE CONCLUSA.":GOT
O310:rem 142
870 PRINT"{GIU}'ERRORE DI CARICAMENTO.RIP
ROVA.{GIU}'":IFDV=1 THEN800:rem 43
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
;E2$:CLOSE15:GOTO800:rem 102
1000 REM CICALINO:rem 231
1001 POKE36878,15:POKE36874,190:rem 206
1002 FORW=1TO300:NEXTW:rem 117
1003 POKE36878,0:POKE36874,0:RETURN
:rem 74
2000 REM CAMPANELLO:rem 130
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368
76,240:NEXTW:rem 22
2002 POKE36876,0:RETURN:rem 119
3000 PRINTC$;"{RVS}NON IN PAGINA ZERO O
{ 2 SPAZI}SU ROM{ 4 GIU}'":GOTO1000

```

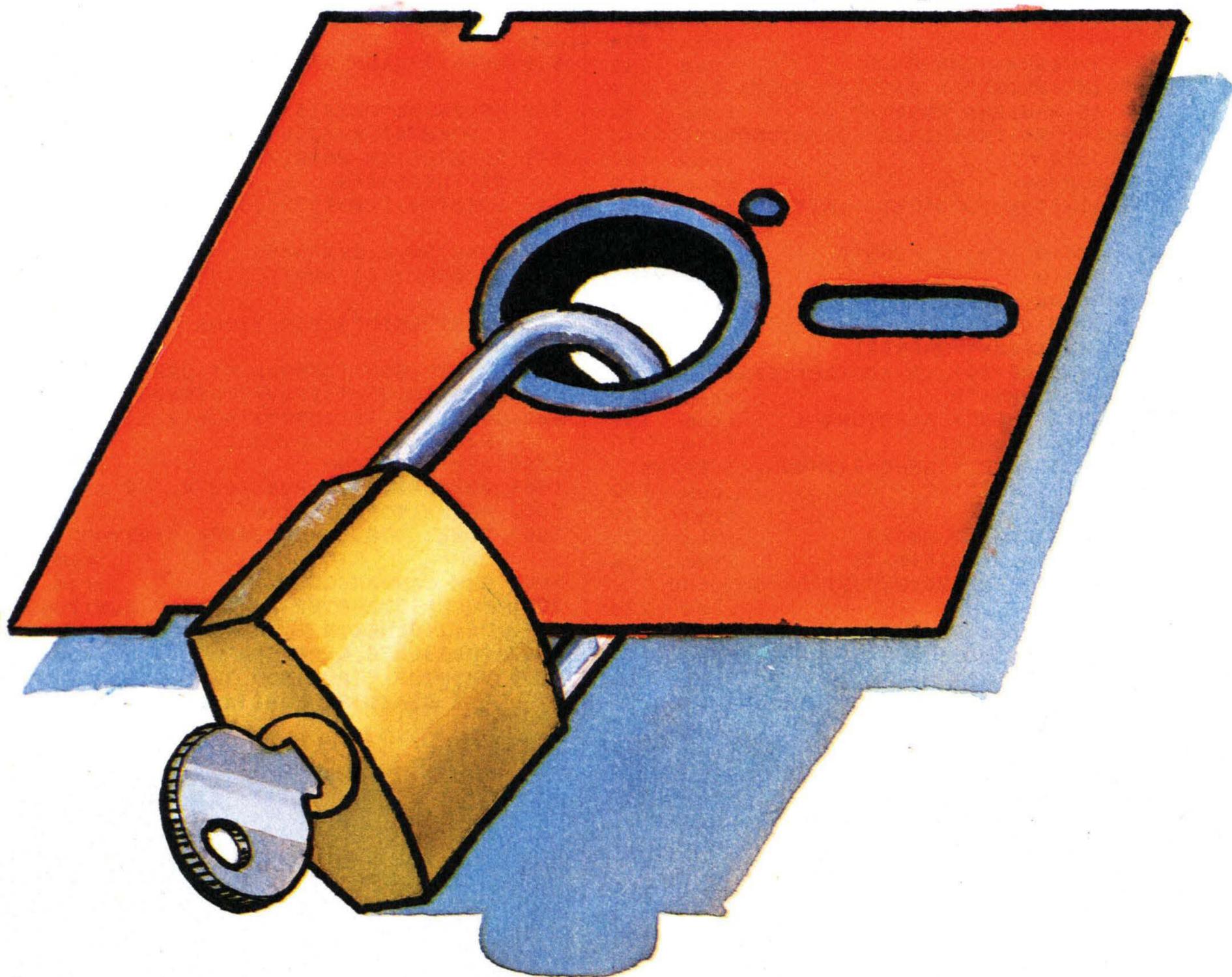


...then



Come proteggere dalla scrittura i propri dischetti

di **K. Hawkins**
trad. e adatt.
di **S. Albarelli**



Chi possiede un'unità singola per floppy disk Commodore, saprà certamente quanto sia facile, per un banale errore di programmazione, perdere i programmi presenti in un dischetto o cancellarne alcuni e non poterli più recuperare. Con questa utility i programmi salvati sui vostri dischetti saranno al sicuro come in una cassaforte e non correranno alcun rischio di essere accidentalmente cancellati.

Tutti i possessori di un'unità per floppy disk sanno che ponendo un pezzo di nastro adesivo, o una delle etichette fornite insieme ai floppy disk, sulla tacchetta presente su un lato del dischetto, si può impedire la sovrascrittura su quel disco.

In tal modo è possibile caricare programmi, ma non salvarne ed è possibile leggere file, ma non scriverne.

Comunque i difetti di questo sistema di protezione sono molti e rilevanti.

Infatti, l'etichettina o il nastro adesivo

sono facilmente asportabili da chiunque (qualsiasi persona competente in fatto di computer potrebbe quindi, all'insaputa del padrone del dischetto, manometterne il contenuto).

Inoltre un'insufficiente aderenza del nastro adesivo o dell'etichetta al dischetto potrebbe rendere difficoltosi l'inserimento e l'estrazione dall'unità lettrice. Questi sono solo due dei difetti che presenta questo sistema, ma ve ne sono innumerevoli altri.

Compatibilità tra unità diverse e Flag del DOS

Il sistema operativo dell'unità a floppy disk, cioè il DOS (Disk Operating System) dell'unità a floppy disk 1541, discende dal DOS delle unità più grandi, utilizzate dai computer Commodore di modello industriale.

La Commodore ha fatto ciò per rendere soddisfacentemente compatibili i floppy disk di tutti i computer prodotti e gestiti dalle relative unità a floppy disk.

Quando un disco viene formattato, viene diviso in tracce e settori.

Il blocco di dati identificato dal settore 0 della traccia 18 contiene la mappa dei blocchi, il nome del disco, l'ID del disco e altre informazioni importanti. Il byte 2 di questo blocco contiene un flag che indica la versione del DOS usata per formattare il dischetto.

Normalmente, esso contiene il valore \$41 (65 decimale, "A" in ASCII).

Questo valore segnala che il dischetto è stato formattato con un'unità 1541 o 4040, che sono tra loro compatibili. Ciò significa che un 1541 può leggere un dischetto formattato con un 4040 e viceversa.

Teoricamente un 1541 potrebbe persino scrivere su un dischetto formattato da un 4040, ma in realtà tale operazione non darebbe buon esito, poiché tra il 1541 e il 4040 c'è una leggera differenza di allineamento delle testine.

C'è inoltre un'unità per floppy disk Commodore, chiamata 2040, che al momento della formattazione fornisce il dischetto di un numero maggiore di settori. Se un dischetto è stato formattato mediante questa unità, contiene nel byte 2 del settore 0 della traccia 18 il valore \$01 (1 decimale).

I dischetti formattati da un 2040 sono compatibili solo in lettura e non in scrittura con il 1541.

Infatti, se si tenta di salvare con un 1541 un programma su un dischetto in formato 2040, si otterrà l'errore numero 73 del DOS, la luce rossa comincerà a lampeggiare e non verrà salvato nulla.

Proteggere un dischetto dalla scrittura

Se si cambiasse il valore del flag presente nel byte 2 del settore 0 della traccia 18, potremmo far credere al computer che il dischetto è stato formattato con un'unità 2040.

Perciò quando si tentasse di salvare qualche programma su un dischetto di cui è stato modificato il valore del flag, il computer, pensando che il dischetto è stato formattato con un 2040, non salverebbe nulla.

Ciò può proteggere il dischetto da sovrascritture e da "scratch", cioè cancellazioni, di file e programmi.

Questo comunque NON protegge i file dalla copiatura.

Esistono in commercio programmi che copiano un intero dischetto in un altro disco.

Utilizzando tali utility per copiare un dischetto protetto in tal modo dalla sovrascrittura, anche la copia sarà protetta, poiché esse lavorano anche sulla traccia 18 e sul suo settore 0, copiando il flag.

Se invece si copia un solo programma, caricandolo e salvandolo in un altro disco, esso, essendo stato salvato su un disco sproteetto, potrà essere cancellato senza problemi (dal dischetto sproteetto, naturalmente).

Dopo aver modificato il flag si potrà leggere file e caricare programmi, ma non sarà possibile salvare programmi, scrivere file, cancellarne, cambiarne il nome, né dare il comando "validate" all'unità.

L'unico comando di scrittura che si può utilizzare è il comando "N: nome del disco, ID", che formatta il dischetto. Utilizzandolo, il disco verrà automaticamente sproteetto, ma verranno persi tutti i programmi in esso contenuti.

Sproteggere un dischetto

Cambiare il contenuto del flag di DOS presente nel byte 2 del settore 0 della traccia 18 è molto semplice da attuare, utilizzando i comandi "Block-Read" (U1) e "Block-Write" (U2).

Come fare se si cambia idea e si vuole nuovamente scrivere su un disco protetto? Mettendo il valore originale nel flag di DOS!

Esso, come già detto, è \$41.

Facendo ciò il processo di protezione si converte.

Però, poiché il dischetto è protetto, non ci è concesso usare questi due comandi e perciò è necessario ricorrere al comando "Memory-Write", molto simile a una POKE, attuata però all'interno della memoria del dischetto.

Facendo ciò, è possibile usare i due comandi sopra indicati per sproteggere il

dischetto, riportando il flag al suo valore consueto.

Come usare il programma "DISK LOCK"

Questo programma è molto facile da usare ed è disponibile nella versione VIC 20 e C16-Plus/4.

Quando lo si fa girare, è necessario inserire nell'unità a floppy disk il dischetto su cui bisogna operare.

All'inizio verrà visualizzata una schermata di presentazione, nella quale, dopo l'apertura di alcuni file per comunicare con il drive, verrà stampato il messaggio: "(P)roteggi o (S)proteggi" seguito da un punto interrogativo.

A questo punto è necessario scegliere se proteggere o sproteggere il dischetto inserito nel drive.

Per comunicare al computer la propria scelta bisognerà poi premere l'iniziale dell'opzione scelta (P per proteggere e S per sproteggere).

Fatto ciò il computer modificherà, in base alla scelta dell'utente, il famoso byte 2 del settore 0 della traccia 18.

Quando avrà finito il computer chiuderà i file e farà girare nuovamente il programma, in caso che l'utente desideri agire su altri dischetti.

L'utente, per comodità, potrà marcare i dischetti protetti scrivendo una P sull'etichetta, utilizzando per farlo una penna ad inchiostro con punta morbida (cioè fatta come quella dei pennarelli), in modo da non danneggiare il dischetto.

L'obiettivo finale di "DISK LOCK" è di proteggere dalla sovrascrittura dischetti di archivi o "backup", cioè copie di sicurezza di dischetti importanti.

È consigliabile non proteggere tutti i propri dischetti, poiché, in caso di necessità, bisognerà ricorrere al programma "DISK LOCK" per poter salvare in essi altri programmi.

N.B. Per tutti coloro che non possiedono la cassetta allegata a questo numero di SuperCommodore ricordo che è molto importante battere attentamente il programma, poiché esso contiene molti comandi complessi diretti al drive, nei quali anche una virgola o un segno di tabulazione errato può creare seri problemi e compromettere la sicurezza dei programmi contenuti nei dischetti protetti mediante l'utility.

"Disk lock"

versione per C16-Plus/4

```
5 GOSUB10000 :rem 166
10 OPEN15,8,15:PRINT#15,"I0":OPEN2,8,2,"#
   " :rem 27
20 PRINT#15,"U1:";2;0;18;0:GOSUB1000
   :rem 10
30 PRINT#15,"B-P:";2;2 :rem 188
40 PRINT"{HOME}{ 17 GIU'}(P)ROTEGGI O (S)
   PROTEGGI" :rem 111
```

```
45 GETA$:IFA$="S"THENPRINT"S":GOTO70
   :rem 221
50 IFA$<>"P"THEN45 :rem 255
55 PRINT"P" :rem 139
60 PRINT#2,CHR$(1);:GOSUB1000:GOTO100
   :rem 177
70 PRINT#2,CHR$(65);:GOSUB1000 :rem 232
80 PRINT#15,"M-W";CHR$(1);CHR$(1);CHR$(1)
   ;CHR$(65) :rem 241
90 PRINT#15,"M-W";CHR$(2);CHR$(7);CHR$(1)
   ;CHR$(65) :rem 249
```

```

100 PRINT#15,"U2:";2;0;18;0:GOSUB1000      :rem 58
110 CLOSE2:PRINT#15,"I0":CLOSE15:FORQ=1TO  :rem 118
    2000:NEXT:RUN
1000 INPUT#15,EN,EM$,ET,ES                  :rem 4
1010 IFENTHENPRINT"{HOME}{ 21 GIU' }"EN",  :rem 210
    ET","ES:PRINT"{GIU' }"EM$
1020 RETURN                                  :rem 163
10000 PRINT"{CLR}"                          :rem 85
10010 PRINT"{RED} U{ 3 *}I UIU{ 4 *}IUI    :rem 245
    U*I"
10020 PRINT"{RED} -U*I- { 3 -}U{ 3 *}K    :rem 11
    { 2 -}UKUK{ 5 SPAZI}{BLK}{RVS}
    { 12 SPAZI}"
10030 PRINT"{RED} { 2 -} -JI{ 3 -}J      :rem 63
    { 3 *}I-JKUK{ 6 SPAZI}{BLK}{RVS}
    {CYN} [<8>][< 2 +>]{BLK}{ 8 SPAZI}
    "
10040 PRINT"{RED} { 2 -} JI{ 3 -}J      :rem 67
    { 3 *}I{ 2 -}UIJI{ 6 SPAZI}{BLK}
    {RVS} [<8>][< 3 +>]{BLK}{ 8 SPAZI}
    "
10050 PRINT"{RED} -J{ 2 *}K{ 3 -}U      :rem 201
    { 3 *}K{ 3 -}JIJI{ 5 SPAZI}{BLK}
    {RVS}{ 4 SPAZI}U{ 2 *}I{ 4 SPAZI}"
10060 PRINT"{RED} J{ 4 *}KJKJ{ 4 *}KJK J  :rem 135
    *K{ 5 SPAZI}{BLK}{RVS}{ 4 SPAZI}-U
    I-{ 4 SPAZI}"
10070 PRINT"{RED}{ 26 SPAZI}{BLK}{RVS}  :rem 214
    { 4 SPAZI}-JK-{ 4 SPAZI}"
10080 PRINT"{RED} UI{ 4 SPAZI}U{ 4 *}IU   :rem 115
    { 4 *}IUI U*I {BLK}{RVS}{ 4 SPAZI}
    J{ 2 *}K{ 4 SPAZI}"
10090 PRINT"{RED} { 2 -}{ 4 SPAZI}-U     :rem 112
    { 2 *}I{ 2 -}U{ 3 *}K{ 2 -}UKUK
    {BLK}{RVS}{ 5 SPAZI}UI{ 5 SPAZI}"
10100 PRINT"{RED} { 2 -}{ 4 SPAZI}      :rem 53
    { 2 -}{ 2 SPAZI}{ 4 -}{ 4 SPAZI}-J
    KUK{ 2 SPAZI}{BLK}{RVS}{ 5 SPAZI}
    { 2 -}{ 5 SPAZI}"
10110 PRINT"{RED} { 2 -}{ 4 SPAZI}      :rem 113
    { 2 -}{ 2 SPAZI}{ 4 -}{ 4 SPAZI}-U
    IJI{ 2 SPAZI}{BLK}{RVS}{ 5 SPAZI}J
    K{ 5 SPAZI}"
10120 PRINT"{RED} -J{ 3 *}I-J{ 2 *}K     :rem 240
    { 2 -}J{ 3 *}I{ 2 -}JIJI"
10130 PRINT"{RED} J{ 4 *}KJ{ 4 *}KJ     :rem 139
    { 4 *}KJK J*K"
10140 PRINT"{ 2 GIU' }"                  :rem 233
10150 PRINT"{ 24 SPAZI}{BLU}{RVS}U      :rem 215
    { 3 *}I"
10160 PRINT"{ 24 SPAZI}{BLU}{RVS}-      :rem 180
    { 3 SPAZI}-"
10170 PRINT"{ 24 SPAZI}{BLU}{RVS}-UI    :rem 98
    [<5>]{ 3 SPAZI}{<*>}E[<*>]E[<*>]
    E[<*>]"
10180 PRINT"{ 24 SPAZI}{BLU}{RVS}-JK    :rem 30
    [<5>]{ 9 SPAZI}{OFF}E"
10190 PRINT"{ 24 SPAZI}{BLU}{RVS}-      :rem 183
    { 3 SPAZI}-"
10200 PRINT"{ 24 SPAZI}{BLU}{RVS}J     :rem 202
    { 3 *}K"
10300 RETURN                              :rem 212

```

"Disk lock"

versione per VIC 20

```

5 GOSUB10000                               :rem 166
10 OPEN15,8,15:PRINT#15,"I0":OPEN2,8,2,"#  :rem 27
    "
20 PRINT#15,"U1:";2;0;18;0:GOSUB1000     :rem 10
10000 PRINT"{CLR}"                          :rem 188
30 PRINT#15,"B-P:";2;2
40 PRINT"{HOME}{ 14 GIU' }{RVS}P{OFF}ROTEG :rem 226
    GI O {RVS}S{OFF}PROTEGGI"
45 GETA$:IFA$="S"THENPRINT"S":GOTO70      :rem 221
50 IFA$<>"P"THEN45                          :rem 255
55 PRINT"P"                                  :rem 139
60 PRINT#2,CHR$(1);:GOSUB1000:GOTO100     :rem 177
70 PRINT#2,CHR$(65);:GOSUB1000             :rem 232
80 PRINT#15,"M-W";CHR$(1);CHR$(1);CHR$(1) :rem 241
    ;CHR$(65)
90 PRINT#15,"M-W";CHR$(2);CHR$(7);CHR$(1) :rem 249
    ;CHR$(65)
100 PRINT#15,"U2:";2;0;18;0:GOSUB1000     :rem 58
110 CLOSE2:PRINT#15,"I0":CLOSE15:FORQ=1TO  :rem 118
    2000:NEXT:RUN
1000 INPUT#15,EN,EM$,ET,ES                  :rem 4
1010 IFENTHENPRINT"{HOME}{ 18 GIU' }"EN",  :rem 159
    ET","ES:PRINT"{GIU' }"EM$
1020 RETURN                                  :rem 163
10000 PRINT"{CLR}":POKE36879,24           :rem 97
10010 PRINT"{ 14 SPAZI}{RVS}{BLK}        :rem 101
    { 7 SPAZI}"
10020 PRINT"{RVS}{RED}{DES} [<*>][<K>]    :rem 21
    {OFF}[<K>]{RVS} [<@>][<K>]E{OFF}
    E{ 4 SPAZI}{RVS}{BLK} {OFF}{PUR}
    [<+>] {RVS}{BLK}{ 4 SPAZI}"
10030 PRINT"{RVS}{RED}{DES}[<M>] [<K>]    :rem 33
    {OFF}[<K>]{RVS}{ 2 SPAZI}[<K>]
    [<*>]{ 4 DES}{BLK}{ 7 SPAZI}"
10040 PRINT"{RVS}{RED}{DES}{ 2 SPAZI}    :rem 32
    [<K>]{OFF}[<K>]{RVS}[<T>] [<K>]
    {OFF}[<*>]{RVS} { 4 DES}{BLK}
    { 2 SPAZI}U*I{ 2 SPAZI}"
10050 PRINT"{ 14 SPAZI}{RVS}{BLK}        :rem 250
    { 2 SPAZI}-W-{ 2 SPAZI}"
10060 PRINT"{RVS}{RED}{ 2 DES} {DES}[<K>] :rem 175
    {OFF}[<K>]{RVS}{ 2 SPAZI}[<K>]E
    {OFF}E{ 2 SPAZI}{RVS}{BLK}
    { 2 SPAZI}J*K{ 2 SPAZI}"
10070 PRINT"{RVS}{RED}{ 2 DES} {DES}[<K>] :rem 189
    -{OFF}[<K>]{RVS} * [<K>] [<*>]
    { 2 DES}{BLK}{ 3 SPAZI}[<I>]
    { 3 SPAZI}"
10080 PRINT"{RVS}{RED}{ 2 DES}{ 2 SPAZI} :rem 35
    [<K>] {OFF}[<K>]{RVS}{ 2 SPAZI}
    [<K>]{OFF}[<*>]{RVS} { 2 DES}{BLK}
    { 3 SPAZI}{DES}{ 3 SPAZI}"
10090 PRINT"{ 14 SPAZI}{RVS}{BLK}        :rem 179
    { 3 SPAZI}{OFF}[<I>]{RVS}
    { 3 SPAZI}"
10100 PRINT"{HOME}";                       :rem 17
10300 RETURN                              :rem 212

```

REM:HW

Hardware

Come utilizzare l'interfaccia RS-232

di M. Cristuib Grizzi

Il VIC 20 ed il Plus/4 sono dotati di interfaccia RS-232, apparato conosciuto da tutti almeno di nome, ma di cui in genere si sa ben poco, anche per le carenze del manuale di istruzioni che accompagna la macchina.

L'interfaccia RS-232 dei computer Commodore non segue, anche se per piccoli dettagli, il vero e proprio standard RS-232 ed è quindi da considerarsi una pseudo RS-232. La differenza rispetto allo standard sta nel livello di tensione in uscita, che oscilla nei Commodore da 0V a +5V, mentre lo standard prevede un intervallo di tensione tra -12V a +12V.

Tale differenza, se si desidera effettuare un collegamento con un apparato esterno diverso da un altro computer Commodore, impone l'utilizzo di apposite cartucce, che ripristinano i livelli di tensione in conformità ai valori standard. La più comune di queste cartucce è prodotta dalla stessa Commodore, ed è denominata VIC-1011A.

L'utilizzo più classico della RS-232 è infatti il collegamento con stampanti non Commodore (noi ad esempio utilizziamo la cartuccia di cui sopra per realizzare i listati della rivista attraverso una grossa stampante Diablo 630), oppure con dei modem, o ancora direttamente con altri computer per il trasferimento di dati o programmi.

La RS-232 è un'interfaccia *seriale*: ciò significa che i bit (sia che costituiscano dati o programmi) vengono trasmessi o ricevuti sequenzialmente, uno dopo l'altro; un byte verrà quindi trasmesso come otto bit in sequenza. Le interfacce *parallele*, al contrario, trasmettono un byte (otto bit) alla volta, e sono quindi in genere molto più veloci delle seriali.

Un termine che si incontra quasi sempre quando si parla di trasmissioni è *buffer*: il più noto tra i buffer esistenti nei com-

puter Commodore è senz'altro il buffer del registratore a cassette, che immagazzina i gruppi di byte da trasferire dal o al registratore. Il meno noto è sicuramente quello dell'interfaccia RS-232, che non è permanente come quello del registratore, ma viene automaticamente allocato in memoria soltanto nel momento dell'apertura di un canale di comunicazione RS-232.

Un canale di comunicazione RS-232 si apre con la consueta istruzione OPEN,

ma con una sequenza di parametri un po' diversa dal solito:

OPEN f,p,s,CHR\$(con)+CHR\$(com)

dove i parametri hanno i seguenti significati:

f: numero del file logico, che può assumere valori da 1 a 255. Valori compresi tra 128 e 255 provocheranno tuttavia l'aggiunta di un *linefeed*, cioè di un CHR\$(10), dopo ogni CHR\$(13), che



rappresenta un carattere di RETURN. Il carattere di *linefeed* viene a volte richiesto dopo un RETURN da certe stampanti che non fanno avanzare di uno spazio la carta dopo ogni fine linea.

p: numero di periferica. L'interfaccia RS-232 viene trattata dal computer come una periferica a sé stante, e il suo numero caratteristico, assegnato dal Sistema Operativo, è 2. Il parametro p dovrà perciò sempre avere il valore 2.

s: indirizzo secondario. Nel caso dell'interfaccia RS-232 non è significativo, e vale perciò 0.

CHR\$(con): carattere di controllo. Questo parametro definisce importanti fattori, come numero dei *bit di stop*, *lunghezza della parola*, e *baud rate*, che definisce la velocità di trasmissione. Questi tre parametri di solito spaventano l'utente, mentre in realtà in essi non vi è proprio nulla di complicato: l'unica cosa veramente importante è che essi siano regolati nello stesso modo sia nell'unità in trasmissione che nell'unità in ricezione. Vedremo più avanti come determinare il valore da assegnare al carattere di controllo.

CHR\$(com): carattere di comando (è tuttavia opzionale). Similmente a quanto effettuato dal carattere di controllo, questo parametro definisce *parità*, *full o half duplex* e *handshaking*. Il metodo di determinazione di questi valori sarà analizzato più avanti.

Alcuni esempi di istruzioni di apertura di un canale RS-232 sono:

```
OPEN1,2,0,CHR$(6)+CHR$(16)
```

```
OPEN2,2,0,CHR$(166)
```

```
OPEN130,2,0,CHR$(166)
```

I dati si ricevono o si trasmettono tramite RS-232 esattamente come si fa abitualmente con i file sequenziali: per ricevere si usa l'istruzione GET #, mentre per trasmettere si usa la PRINT #. Alla fine della comunicazione si chiuderà il canale con la consueta CLOSE f. Il semplice comando:

```
OPEN1,2,0,CHR$(6):CMD1:LIST
```

seguito, alla ricomparsa del cursore, da:

```
PRINT # 1:CLOSE1
```

invia su RS-232 il listato del programma che si trova in memoria, ed è - tranne che per la forma della OPEN - assolutamente analogo al comando che si usa per effettuare un listato su stampante, nastro o disco.

Il carattere di controllo

Come abbiamo detto, il carattere di controllo definisce rispettivamente bit di stop, lunghezza della parola e baud rate. I (o il) bit di stop servono per aiutare l'unità ricevente a stabilire quando è arrivata la fine di un carattere trasmesso (ricordate che ogni carattere viene inviato come sequenza di bit); è possibi-

le usare uno o due bit di stop.

Il carattere di controllo, come del resto qualsiasi altro carattere ASCII Commodore, è composto da otto bit. Immaginandolo con tutti i bit posti a zero, esso avrà la seguente forma:

```
bit          7 6 5 4 3 2 1 0
carattere di controllo 0 0 0 0 0 0 0 0
```

Il numero di bit di stop dipende dal valore che decidiamo di assegnare al bit 7 del carattere di controllo: se lo poniamo a uno, ci saranno due bit di stop, mentre se lo poniamo a zero ce ne sarà uno solo.

La lunghezza della parola stabilisce invece quanti bit devono essere trasmessi per ogni carattere, e può assumere valori compresi tra cinque e otto bit per carattere, anche se quasi sempre viene usata di otto bit. La lunghezza della parola è definita dal valore assunto dai bit 6 e 5 del carattere di controllo, secondo la seguente tabella:

bit	6	5	lunghezza della parola
	0	0	8 bit
	0	1	7 bit
	1	0	6 bit
	1	1	5 bit

Se volessimo ad esempio definire 2 bit di stop con una lunghezza della parola di sette bit, il carattere di controllo inizierà ad assumere la forma:

```
bit          7 6 5 4 3 2 1 0
carattere di controllo 1 0 1 0 0 0 0 0
```

dove abbiamo posto il bit 7 a 1 (2 bit di stop), e i bit 6 e 5 rispettivamente a 0 ed 1 (lunghezza della parola di sette bit).

Il baud rate stabilisce, come detto, la velocità di trasmissione. Quasi sempre vengono usati valori di 300 o 1200 baud, equivalenti rispettivamente a circa 30 caratteri al secondo e circa 120 caratteri al secondo. Il baud rate si regola attraverso i bit 3, 2, 1 e 0 del carattere di controllo, secondo la seguente tabella:

bit	3	2	1	0	baud rate
	0	0	0	1	50
	0	0	1	0	75
	0	0	1	1	110
	0	1	0	0	134.5
	0	1	0	1	150
	0	1	1	0	300
	0	1	1	1	600
	1	0	0	0	1200

Ritornando all'esempio precedente, se desiderassimo definire 2 bit di stop, lunghezza della parola di sette bit e velocità di trasmissione di 300 baud, il carattere di controllo assumerà la forma:

```
bit          7 6 5 4 3 2 1 0
carattere di controllo 1 0 1 0 0 1 1 0
```

Come avrete notato, il bit 4 del caratte-

re di controllo non viene usato, e rimane quindi sempre a zero.

Non resterà ora altro che convertire il valore ottenuto - 10100110 - da base binaria in base decimale. Dal momento che disponiamo di un computer, deleghiamo ad esso questo "ingrato" compito, attraverso il seguente programma:

```
10 INPUT"VALORE BINARIO";R$
20 IFLEN(R$)<>8THEN!0
30 FORI=0TO7:A=VAL(MID$(R$,L
EN(R$)-I,1))
40 B=B+A*2!I:NEXT I
50 PRINT"VALORE DECIMALE:";B
60 END
```

Otterremo il valore decimale 166: il registro di controllo verrà quindi inserito nell'istruzione OPEN come CHR\$(166).

Un altro valore in genere molto usato del carattere di controllo è CHR\$(6), che corrisponde a 1 bit di stop, lunghezza della parola di 8 bit e 300 baud.

Il carattere di comando

Mentre il carattere di controllo regola bit di stop, lunghezza della parola e baud rate, il carattere di comando regola parità, half o full duplex e handshaking.

La parità viene usata per il controllo degli errori eventualmente verificatisi durante la trasmissione, in genere dovuti a dispersioni o interferenze sulla linea (sia essa una linea telefonica oppure un semplice cavo seriale).

Se si decide di utilizzare il controllo di parità (esso infatti, come tutti i parametri regolati dal carattere di comando, è opzionale), si può scegliere tra parità pari e parità dispari. Le due cose sono assolutamente equivalenti, e l'unico fattore importante è che entrambe le unità in comunicazione siano regolate sullo stesso tipo di parità.

I bit 7, 6 e 5 del carattere di comando definiscono il valore della parità secondo il seguente schema:

bit	7	6	5	parità
	0	0	0	nessuna
	0	0	1	dispari
	0	1	1	pari

Quando si usa una lunghezza della parola di otto bit, l'opzione di parità non è tuttavia disponibile ed occorre quindi porre a zero i bit 7, 6 e 5 del carattere di comando (nessuna parità).

Supponendo invece di scegliere una parità dispari, il carattere di comando assumerà la forma:

```
bit          7 6 5 4 3 2 1 0
carattere di comando 0 0 1 0 0 0 0 0
```

Il tipo di duplex indica se entrambe le unità in collegamento possono trasmettere simultaneamente (full duplex), oppure se la trasmissione deve avvenire da parte di un'unità alla volta (half du-

REM=HW

plex). Il valore senz'altro generalmente più usato è il full duplex, tuttavia anche qui è importante che entrambe le unità siano regolate allo stesso modo: o entrambe sul full duplex, oppure entrambe su half duplex.

Il bit 4 del carattere di comando regola il tipo di duplex:

bit	4	duplex
	0	full
	1	half

L'handshaking è regolato dal bit 0 del carattere di comando e indica come le unità riceventi e trasmettenti riconoscono e stabiliscono la comunicazione tra di esse (protocollo di comunicazione). Per la maggior parte dei casi si usa un valore 0 per questo bit.

Volendo regolare parità dispari e half duplex, il carattere di comando avrà la forma:

bit	7	6	5	4	3	2	1	0
carattere di comando	0	0	1	1	0	0	0	0

I bit 3, 2 e 1 del carattere di comando non sono usati e vanno quindi lasciati a zero. Con il precedente programmino potremo convertire in decimale anche questo valore, ottenendo per l'ultimo esempio un CHR\$(48).

La sintassi per l'apertura di un canale RS-232 dotato di 2 bit di stop, lunghezza della parola di 7 bit, 300 baud, parità dispari ed half duplex sarà dunque:

```
OPEN1,2,0,CHR$(166)+CHR$(48)
```

Come avete visto, il carattere di comando non è così importante come il carattere di controllo, e viene infatti quasi sempre omesso per la maggior parte degli scopi.

Il controllo degli errori

Quando viene eseguita una OPEN sulla RS-232, come detto viene allocato in memoria un buffer di comunicazione, lungo 512 byte (256 byte per l'INPUT e 256 byte per l'OUTPUT).

Viene inoltre eseguito automaticamente un CLR, che cancella il contenuto di tutte le variabili, chiude tutti i file aperti e fa "dimenticare" al computer la linea di ritorno di un'istruzione RETURN eventualmente incontrata dopo la OPEN. Per questo motivo un canale RS-232 non deve mai venire aperto in una subroutine, oppure dopo che siano stati dimensionati vettori o matrici o assegnate variabili importanti: è sempre opportuno aprire il canale RS-232 con la prima linea del programma.

Anche la CLOSE di un canale RS-232 provoca un analogo CLR, e deve quindi preferibilmente essere usata come ultima istruzione del programma.

Come ben sa chi utilizza abitualmente i file sequenziali su nastro o disco, la variabile riservata ST (STatus) controlla lo stato del canale di comunicazione. Nel caso della RS-232 essa assume tuttavia dei significati diversi dal solito: in

particolare i suoi bit 0, 1 e 2 indicano tre diversi tipi di errore:

bit	0	1	2	tipo errore
0	0	0	0	tutto OK
1	0	0	0	err.parità
0	1	0	0	err.pacchetto
0	0	0	1	err.overflow

Ovviamente, gli errori possono verificarsi contemporaneamente: ad esempio, se i bit 0 e 2 sono entrambi a 1, si sono verificati un errore di parità e un overflow nel buffer dell'unità ricevente.

Un errore di parità si verifica se un certo bit ha un valore non congruente con il controllo di parità (interferenze sulla linea). Si ricordi di non considerare significativo questo bit, se si utilizza una lunghezza della parola di otto bit.

Un errore di pacchetto si verifica quando l'unità ricevente ha problemi nel ricevere un "pacchetto" di bit indicante un carattere.

Un errore di overflow, infine, indica che il buffer di INPUT dell'unità ricevente è pieno (256 byte) e viene ricevuto il 257esimo carattere prima che il programma di comunicazione abbia iniziato a svuotare il buffer. Ciò accade tipicamente quando il programma - generalmente in BASIC - è troppo lento nella sua esecuzione rispetto alla velocità di trasmissione selezionata: i dati arrivano più in fretta di quanto il programma sia in grado di processarli. Questi bit della variabile riservata ST si possono controllare da BASIC attraverso un AND logico; ad esempio:

```
IFST AND1THENPRINT"ERR.PARITA'"
```

```
IFST AND2THENPRINT"ERR.PACCHETTO"
```

```
IFST AND4THENPRINT"ERR.OVERFLOW"
```

Si noti lo spazio tra ST e AND, dovuto ad un "bug" nell'interprete BASIC, che segnala un "SYNTAX ERROR" nel caso tale spazio sia assente.

Infine, un'ultima nota per quanto concerne la variabile ST: essa viene posta a zero dal computer dopo l'esecuzione di un'istruzione di INPUT/OUTPUT come GET # o PRINT #. Il solito truccetto per aggirare questo inconveniente è definire una variabile di lavoro uguale alla ST:

```
S=ST:GET # 1,A$:IFSAND1THEN...
```

Pensiamo con ciò di aver esaurito l'aspetto pratico dell'utilizzo dell'interfaccia RS-232: con la sempre maggiore diffusione di periferiche come i modem, non guasta infatti iniziare a prendere un po' di confidenza con questa poco conosciuta caratteristica del nostro computer.

Il programmino presentato in fondo all'articolo permette di inviare a un'unità ricevente (modem, stampante, altro computer...) un file sequenziale su disco. Esso richiede tutti i parametri di comunicazione analizzati nell'articolo, quindi calcola e visualizza i caratteri di controllo e di comando per la OPEN. Richiederà quindi il nome del file sequenziale (che può essere un testo di un word processor, un listato di un programma BASIC, un file dati, ecc...) e inizierà a trasmetterlo su RS-232, visualizzandolo contemporaneamente su video. Il programma funziona su VIC 20 e Plus/4.



Invio di file su RS-232 per VIC 20 e Plus/4

```

10 INPUT"{CLR}BIT DI STOP{ 2 SPAZI}1
   { 3 SIN}";A$:IFA$="1"THENR$="0":GOTO30
   :rem 46
20 R$="1" :rem 138
30 INPUT"LUNGH.PAROLA{ 2 SPAZI}7{ 3 SIN}
   ";A$:IFA$="7"THENR$=R$+"01" :rem 94
40 IFA$="8"THENR$=R$+"00" :rem 56
50 IFA$="6"THENR$=R$+"10" :rem 56
60 IFA$="5"THENR$=R$+"11" :rem 57
70 R$=R$+"0" :rem 47
80 INPUT"BAUD RATE{ 3 SPAZI}300{ 5 SIN}";
   B :rem 165
90 IFB=300THENR$=R$+"0110" :rem 147
100 IFB=1200THENR$=R$+"1000" :rem 234
110 IFB=2400THENR$=R$+"1010" :rem 239
120 IFB<>300ANDB<>1200ANDB<>2400THEN80
   :rem 175
130 B=0:FORI=0TO7:A=VAL(MID$(R$,LEN(R$)-I
   ,1)): B=B+A*2↑I:NEXT :rem 45
140 W1$=CHR$(B):PRINT"{GIU'}CHR$("B")
   {GIU'}" :rem 23
150 R$="":PRINT"PARITA'(1-NO/2-DISP/3-PAR
   I)":INPUT"{ 2 SPAZI}1{ 3 SIN}";P
   :rem 32
160 IFP=1THENR$="000" :rem 154
170 IFP=2THENR$="001" :rem 157
180 IFP=3THENR$="011" :rem 160
190 INPUT"0-FULL.D/ 1-HALF.D{ 2 SPAZI}0
   { 3 SIN}";A:IFA=0THENR$=R$+"0":GOTO21
   0 :rem 129
200 R$=R$+"1" :rem 91
210 R$=R$+"000" :rem 187
220 INPUT"HANDSHAKE (0-3/1-X){ 2 SPAZI}0
   { 3 SIN}";A:IFA=0THENR$=R$+"0":GOTO24
   0 :rem 175
230 R$=R$+"1" :rem 94
240 B=0:FORI=0TO7:A=VAL(MID$(R$,LEN(R$)-I
   ,1)):B=B+A*2↑I:NEXT :rem 47
250 W2$=CHR$(B):PRINT"{GIU'}+CHR$("B")
   {GIU'}" :rem 69
251 PRINT"{GIU'}PREMI UN TASTO" :rem 37
252 GETA$:IFA$=""THEN252 :rem 85
300 OPEN2,2,0,W1$+W2$ :rem 97
310 OPEN15,8,15,"I" :rem 219
320 PRINTCHR$(14)CHR$(8) :rem 99
330 PRINTCHR$(147)CHR$(17)CHR$(17)"NOME F
   ILE":INPUT FI$ :rem 149
340 OPEN3,8,3,FI$+",S,R" :rem 10
350 INPUT#15,EN,EM$,ET,ES:IFENTHENPRINTEM
   $:CLOSE3:CLOSE2:CLOSE15:END :rem 149
360 GET#3,A$:RS= ST :IFA$=""THENPRINT"FIL
   E NON CORRETTO":STOP:A$=CHR$(0)
   :rem 45
370 PRINTA$;:PRINT#2,A$; :rem 175
380 IFRS=0THEN360 :rem 4
390 PRINT#2,CHR$(A);CHR$(0) :rem 250
400 INPUT#15,EN,EM$,ET,ES:IFENTHENPRINTEM
   $:CLOSE3:CLOSE2:CLOSE15:END :rem 145
410 CLOSE15 :rem 113
420 FORI=0TO255:PRINT#2,CHR$(0);:NEXT
   :rem 240
430 PRINT"UN ALTRO FILE ?"; :rem 39
440 GETA$:IFA$=""THEN440 :rem 83
450 IFA$="S"THENCLR:GOTO330 :rem 122
460 CLOSE2:END :rem 83

```



Dalla grande edicola Jackson Tutto sull'hobby e home computer

**VIDEO
Giochi
& COMPUTER**

La guida indiscussa al fantastico mondo dei videogames. La più eccitante, divertente, istruttiva rassegna del settore.
11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 33.000

**strumenti
MUSICALI**

Il mondo delle 7 note in versione ... elettronica. Con test strumentali, novità e analisi del mercato, servizi speciali.
11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

**ELETRONICA
HOBBY**

Il nuovo fai da te per l'hobbista elettronico, con progetti sempre nuovi e originali da realizzare e divagazioni sul mondo dei micro computer.
12 numeri all'anno: L. 5.500 a numero
Abbonamento: L. 32.000

Quando l'informazione fa testo

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

Desidero ricevere GRATIS un numero
della Rivista _____

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

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

Nome _____

Cognome _____

via _____

CAP _____ Città _____



House of terror per VIC 20



di M. Marinai

In questo gioco impersonate un uomo alla ricerca della ricchezza. Dopo lunghe indagini avete scoperto che una strega tiene nascosto nella sua dimora un tesoro di inestimabile valore.

Senza pensarci due volte vi recate subito sul posto, e non appena la strega è uscita per la sua solita passeggiata sulla scopa vi introducete furtivamente nella sua casa. Ma nella frenesia di impossessarvi del tesoro avete dimenticato che, come in tutte le case

stregate che si rispettino, anche nella casa in cui vi trovate ci sono trabocchetti e pericoli di ogni genere. Non avendo portato con voi nessun attrezzo o mezzo di difesa, dovrete contare solo sulla vostra agilità nel correre e nel saltare. Riuscirete a raggiungere il 17° piano per impossessarvi del tesoro, evitando gli ostacoli, prima che la strega ritorni?

Le regole del gioco sono poche e semplici. Tutto ciò che dovete fare è... saltare voragini, evitare lampadari che cadono, scansare mostri e palle incantate e altro ancora! Per saltare nell'ultima di-

rezione in cui vi siete mossi premete il pulsante di fuoco. Per passare al piano superiore, quando siete sotto la scala, muovete il joystick in su. Per spostarvi muovete il joystick orizzontalmente. Anche il tempo vi è nemico; infatti, avete solo nove minuti per raggiungere il tesoro prima che la strega vi scopra (in questo caso per voi sarà la fine). Se raggiungete il tesoro, il gioco riprende dall'inizio, ma avrete un minuto di tempo in meno. HOUSE OF TERROR gira su VIC20 con almeno 16 Kbyte di espansione e richiede l'uso del joystick. I tre pro-

grammi di seguito listati vanno digitati e salvati uno dopo l'altro sulla stessa cassetta, mantenendone l'ordine. È importante che non si dia il RUN a nessuno dei tre programmi prima di averli

salvati, pena l'autocancellazione. La seconda e la terza parte del listato devono essere chiamate rispettivamente "HOUSE2" e "HOUSE3". Se intendete utilizzare il disk drive, modificate la li-

nea 60 del programma 1 e la linea 40 del programma 2, come spiegato nelle REM. Non mi resta che augurarvi di diventare miliardari con il tesoro di HOUSE OF TERROR!

House of terror per VIC 20

Programma 1

```

10 REM*****          :rem 187
20 REM*   HOUSE      * :rem 30
30 REM*   OF TERROR  * :rem 14
40 REM*****          :rem 190
50 POKE43,1:POKE44,32:POKE8192,0:POKE3686
   9,240:POKE36866,150:POKE648,30
                               :rem 96
60 A$="LO"+CHR$(34)+"HOUSE2"+CHR$(34)+", "
   :D=1:REM D=8 PER DISK DRIVE :rem 62
70 PRINT"{CLR}{ 2 GIU' }"A$D"{HOME}";:POKE
   198,5:POKE631,13:POKE632,82:POKE633,85
   :POKE634,78:POKE635,13      :rem 151
80 NEW                          :rem 82

```

Programma 2

```

10 PRINT"{CLR}CARICAMENTO{ 2 SPAZI}CARATT
   ERI":FORT=5120TO5631:P=PEEK(T+27648):P
   OKET,PORP*2:NEXT           :rem 60
20 FORT=5632TO6143:POKET,PEEK(T+27648):NE
   XT:FORT=6144TO6463:READA:POKET,A:NEXT
                               :rem 174
30 FORT=6464TO7167:POKET,PEEK(T+27648):NE
   XT                          :rem 137
40 A$="LO"+CHR$(34)+"HOUSE3"+CHR$(34)+", "
   :D=1:REM D=8 PER DISK DRIVE :rem 61
50 PRINT"{CLR}{ 2 GIU' }"A$D"{HOME}";:POKE
   198,5:POKE631,13:POKE632,82:POKE633,85
   :POKE634,78:POKE635,13      :rem 149
60 NEW                          :rem 80
70 DATA0,0,234,170,234,138,142,0,0,0,151,
   210,178,146,146,0,64,0,64,80,64,80,64,
   0                               :rem 106
80 DATA0,0,238,72,78,72,78,0,0,0,151,245,
   151,148,148,0,0,0,112,84,80,84,112,0
                               :rem 222
90 DATA255,255,255,255,255,255,255,255,19
   5,255,255,195,195,255,255,195:rem 227
100 DATA4,14,126,126,4,14,30,62,62,62,62,
   62,12,12,60,124,32,112,126,126,32,112
   ,120,124                       :rem 127
110 DATA124,124,124,124,48,48,60,62,0,0,2
   8,62,65,62,28,0,16,56,124,214,254,198
   ,254,170                       :rem 177
120 DATA16,16,56,124,254,254,16,0,0,0,1,7
   ,15,31,63,127,24,126,255,255,252,252,
   252,252                       :rem 110
130 DATA0,0,128,224,112,24,124,254,127,25
   5,255,241,252,241,252,241      :rem 202
140 DATA248,240,224,224,224,0,224,224,254
   ,255,255,255,255,3,255,254     :rem 20
150 DATA127,63,31,15,7,1,0,0,224,251,249,
   255,255,255,126,24,254,252,248,240,22
   4,128,0,0                       :rem 214
160 DATA0,56,120,56,16,56,108,108,108,108
   ,124,56,16,16,16,48            :rem 183
170 DATA0,56,60,56,16,56,108,108,108,108,
   124,56,16,16,16,24            :rem 133

```

```

180 DATA0,56,186,186,146,186,254,124,124,
   124,124,56,40,40,40,40        :rem 76
190 DATA124,130,124,0,0,56,56,56,16,186,2
   54,254,254,254,254,186        :rem 87
200 DATA255,255,255,255,255,255,255,255,4
   0,40,40,40,0,0,0,0           :rem 112
210 DATA0,7,15,31,63,127,255,128,0,255,25
   5,255,255,255,255,0           :rem 180
220 DATA0,252,252,244,236,212,172,212,157
   ,149,149,149,149,157,128,255:rem 134
230 DATA220,84,212,148,84,92,0,255,172,21
   2,172,212,168,208,160,192     :rem 226

```

Programma 3

```

10 DIMN1%(77),D1%(77),N2%(77),D2%(77),N3%
   (12),D3%(12),N4%(5),D4%(5),N5(46)
                               :rem 77
20 FORT=0TO62:READD1%(T):NEXT:FORT=0TO77:
   READD2%(T):NEXT             :rem 37
30 FORT=0TO64:READN1%(T):NEXT:FORT=0TO77:
   READN2%(T):NEXT             :rem 60
40 FORT=0TO12:READN3%(T),D3%(T):NEXT:FORT
   =0TO5:READN4%(T),D4%(T):NEXT :rem 220
50 FORT=0TO46:READN5(T):NEXT:DEFFNA(Q)=IN
   T(RND(1)*Q)                 :rem 235
60 V=36878:V1=V-4:V2=V-3:V3=V-2:V4=V-1:JJ
   =37154:PA=JJ-3:PB=PA+1:C%=30720
                               :rem 185
70 C$="{WHT}{RED}{CYN}{PUR}{GRN}{BLU}
   {YEL}{WHT}":FORT=0TO7:HI(T)=0:HI$(T)="
   { 10 SPAZI}":NEXT          :rem 178
80 SC=0:T$="000900":L%=1       :rem 224
90 POKEV+1,8:POKEV,15:POKEV-9,253:S%=1:PR
   INT"{CLR}{RVS}{WHT}@AB"TAB(13)"{GRN}CD
   E":IFL%=6THEN2600           :rem 224
100 FORT=7702TO8185STEP22:POKET+C%,6:POKE
   T+C%+21,6:POKET,160:POKET+21,160:NEXT
                               :rem 235
110 FORT=7702TO7723:FORI=0TO462STEP154:PO
   KET+C%+I,6:POKET+I,160:NEXT:NEXT
                               :rem 72
120 FORT=7721TO7809STEP22:POKET+C%,5:POKE
   T,135:NEXT:FORT=7858TO7946STEP22
                               :rem 78
130 POKET+C%,5:POKET,135:NEXT:FORT=8029TO
   8117STEP22:POKET+C%,5:POKET,135:NEXT
                               :rem 75
140 ONL%GOSUB2000,2100,2200,2300,2400:IFL
   %=1THENGOSUB2500:TI$="000000"
                               :rem 97
150 PO%=8143:DS%=1:S%=1:POKEPO%+C%,1:POKE
   PO%-22+C%,1:POKEPO%-22,154:POKEPO%,15
   5                               :rem 46
160 J1%=PEEK(PA):POKEJJ,127:J2%=PEEK(PB):
   POKEJJ,255:F%=-((J1%AND32)=0)
                               :rem 25
170 UP%=((J1%AND4)=0):LF%=((J1%AND16)=0):
   RG%=-((J2%AND128)=0)       :rem 236
180 IFUP%ANDPEEK(PO%-44)=135THEN2700
                               :rem 98

```

```

190 IFF%THEN2800 :rem 141
200 D%=LF%+RG%:IFPEEK(PO%+D%)=160ORD%=0TH
EN250 :rem 128
210 DS%=D%:POKEPO%-22,32:POKEPO%,32:PO%=P
O%+D%:P%=PEEK(PO%):PP%=PEEK(PO%-22)
:rem 194
220 POKEPO%-22+C%,1:POKEPO%+C%,1:POKEPO%-
22,153+DS%:POKEPO%,154+DS% :rem 105
230 IFP%<>32ORPP%<>32ORPEEK(PO%+22)=32THE
N2900 :rem 195
240 POKEV3,200:POKEV3,0 :rem 90
250 IFPEEK(PO%+22)=32THEN2900 :rem 139
255 PRINT"{HOME}"TAB(3)"{WHT}"SCTAB(16)"
{GRN}"TI$:IFTI$>=T$THEN3000 :rem 219
260 ONL%GOTO3300,3700,4100,4500,4900,5300
:rem 110
270 PRINT"{CLR}{GRN}{ 6 SPAZI}FINE GIOCO"
:FORT=0TO12:POKEV2,N3%(T):FORI=0TOD3%
(T):NEXT:NEXT :rem 171
280 POKEV2,0:FORT=0TO2500:NEXT:H=9
:rem 106
290 FORT=7TO0STEP-1:IFSC>HI(T)THENH=T
:rem 146
300 NEXT:IFH=9THEN1000 :rem 73
310 IFH=7THEN330 :rem 164
320 FORT=7TOH+1STEP-1:HI(T)=HI(T-1):HI$(T
)=HI$(T-1):NEXT :rem 104
330 PRINT"{CLR}{WHT}SEI"H+1"{SIN}[<V>] IN
CLASSIFICA !":FORT=0TO5:POKEV2,N4%(T
):POKEV3,N4%(T) :rem 58
340 FORI=0TOD4%(T):NEXT:NEXT:POKEV2,0:POK
EV3,0:PRINT"{GRN}{ 2 GIU'}DIGITA IL T
UO NOME:{ 2 GIU'}{WHT}" :rem 192
350 HI$(H)="" :INPUTHI$(H):IFHI$(H)=""THEN
HI$(H)="ANONIMO" :rem 38
360 HI$(H)=LEFT$(HI$(H)+"{ 10 SPAZI}",10)
:HI(H)=SC :rem 69
1000 PRINT"{CLR}{YEL}{ 3 SPAZI}HOUSE OF
{ 2 SPAZI}TERROR":PRINT"{GIU'}{PUR}
{ 6 SPAZI}CLASSIFICA{GIU'}":FORT=0TO
7 :rem 17
1010 H%=LEN(STR$(HI(T)))-1:H$=RIGHT$(STR$(
HI(T)),H%) :rem 177
1020 PRINTMID$(C$,T+1,1)RIGHT$("0000000000
"+H$,9)" + "HI$(T):NEXT:POKEV,10
:rem 228
1025 PRINT"{ 2 GIU'}{GRN}{ 6 SPAZI}PREMI
FIRE"; :rem 183
1030 X%=0:Y%=0:D1%=D1%(0):D2%=D2%(0)
:rem 45
1040 IFD1%<=0THENX%=X%+1:D1%=D1%(X%)
:rem 210
1050 IFD2%<=0THENY%=Y%+1:D2%=D2%(Y%)
:rem 217
1060 POKEV2,N1%(X%):POKEV3,N2%(Y%):D1%=D1
%-1:D2%=D2%-1:IFD2%(X%)=-1THEN1080
:rem 135
1070 IF(PEEK(PA)AND32)<>0THEN1040
:rem 133
1075 POKEV2,0:POKEV3,0:GOTO80 :rem 9
1080 POKEV2,0:POKEV3,0:FORT=0TO500:IF(PEE
K(PA)AND32)<>0THENNEXT:GOTO1030
:rem 236
1090 T=500:NEXT:GOTO80 :rem 68
2000 PRINT"{HOME}{ 22 GIU'}{ 2 DES}
{ 2 SPAZI}{DES}{ 2 DES}{ 2 SPAZI}
{ 3 DES}{DES}{HOME}" :rem 52
2010 PRINT"{HOME}{ 7 GIU'}{RVS}{BLU}
{ 5 DES}F{DES}FF{ 2 DES}FF{ 2 DES}F
{DES}F":M%=8020:DI%=-1:I%=1:RETURN
:rem 226
2100 FORT=0TO6:LA%(T)=7883+T*2:POKELA%(T)
+C%,7:POKELA%(T),142:NEXT :rem 121
2110 PRINT"{HOME}{ 16 GIU'}{RVS}{BLU}
{ 4 DES}FFF{ 5 DES}FFF":M1%=8058:M2%
=8066 :rem 112
2111 POKEM1%+C%,6:POKEM1%+C%+1,6:POKEM1%+
C%+2,6:POKEM2%+C%,6:POKEM2%+C%+1,6
:rem 167
2112 POKEM2%+C%+2,6 :rem 235
2120 U1%=7708:U2%=7718:W1%=7862:W2%=7872:
I%=1:DI%=-1:Z%=0:RETURN :rem 82
2200 FORT=7861TO7871:POKET,32:NEXT:FORT=8
016TO8026:POKET,32:NEXT :rem 14
2210 PRINT"{HOME}{ 22 GIU'}{ 3 DES}
{ 2 DES}{ 2 SPAZI}{ 3 DES}{ 2 DES}
{HOME}":M1%=7872:I1%=-1:M2%=8019:I2
%=-1 :rem 28
2220 RETURN :rem 166
2300 B1%=8162:I1%=-1:B2%=7994:I2%=1:B3%=7
997:I3%=-1:B4%=7841:I4%=-1:RETURN
:rem 242
2400 M1%=8162:M2%=7835:M3%=7854 :rem 93
2410 POKEM1%+C%,2:POKEM1%-22+C%,2:POKEM2%
+C%,4:POKEM2%-22+C%,4:POKEM3%+C%,5
:rem 171
2420 POKEM3%-22+C%,5:POKEM1%,137:POKEM2%,
139:POKEM3%,137:POKEM1%-22,136:POKEM
2%-22,138 :rem 104
2430 POKEM3%-22,136:FA%(0)=7905:FA%(1)=79
30:FA%(2)=7955:FA%(3)=7980:FA%(4)=79
39 :rem 104
2440 FORT=0TO4:DF%(T)=22:NEXT:Z1%=0:Z2%=0
:Z3%=0:RETURN :rem 19
2500 A%=230:B%=225:FORT=0TO1000:NEXT
:rem 53
2510 FORT=1TO3:POKEV1,A%:POKEV2,A%:POKEV3
,A%:FORI=0TO70:NEXT:POKEV1,0:POKEV2,
0:POKEV3,0 :rem 17
2520 FORI=0TO50:NEXT:NEXT:POKEV1,B%:POKEV
2,B%:POKEV3,B%:FORT=0TO1400:NEXT
:rem 175
2530 POKEV1,0:POKEV2,0:POKEV3,0:IFA%=230T
HENA%=228:B%=221:FORT=0TO1600:NEXT:G
OTO2510 :rem 89
2540 RETURN :rem 171
2600 FORT=0TO2:FORI=0TO2:POKE7726+C%+T*22
+I,1:POKE7726+T*22+I,143+T*3+I:NEXT:
NEXT :rem 103
2610 FORT=0TO10:POKE7778+C%+T*21,2:POKE77
79+C%+T*23,2:POKE7778+T*21,78
:rem 235
2620 POKE7779+T*23,77:NEXT:FORT=8164TO818
5:POKET+C%,6:POKET+C%-154,6:POKET,16
0 :rem 161
2630 POKET-154,160:NEXT:FORT=8010TO8164ST
EP22:POKET+C%,6:POKET+C%+21,6:POKET,
160 :rem 84
2640 POKET+21,160:NEXT:FORT=0TO1:FORI=0TO
2:POKE7969+C%+T*22+I,7 :rem 58
2645 POKE7969+T*22+I,162+T*3+I :rem 117
2650 NEXT:NEXT:FORT=0TO2:LA%(T)=8044+T*2:
POKELA%(T)+C%,7:POKELA%(T),142:NEXT:
Z%=0 :rem 141
2660 FORT=8167TO8171:POKET,32:NEXT:M%=816
9:I1%=1:B%=8153:I2%=-1:FORT=8029TO81
17STEP22 :rem 28
2670 POKET+C%,5:POKET,135:NEXT:GOTO150
:rem 74

```

```

2700 FORT=PO%TOPO%-110STEP-22:POKET,32:NE
XT:POKEPO%-132+C%,6:POKEPO%-132,160
:rem 87
2710 S%=S%+1:IFS%=4THEN3100 :rem 248
2720 IFS%=2ANDL%=6THEN3200 :rem 236
2730 PO%=PO%-154:POKEPO%-22+C%,1:POKEPO%+
C%,1:DS%=1:IFS%=2THENDS%=-1:rem 189
2740 POKEPO%-22,153+DS%:POKEPO%,154+DS%:P
OKEV3,220:POKEV3,0:SC=SC+FNA(1000):G
OTO160 :rem 150
2800 IFPEEK(PO%-44)<>32ORPEEK(PO%-66)<>32
ORPEEK(PO%+DS%)=160THEN250 :rem 182
2810 POKEPO%,32:POKEPO%-22,32:PO%=PO%-22:
P%=PEEK(PO%):PP%=PEEK(PO%-22)
:rem 91
2820 POKEPO%-22+C%,1:POKEPO%-22,153+DS%:P
OKEPO%,154+DS%:IFP%<>32ORPP%<>32THEN
2900 :rem 224
2830 Q%=0 :rem 176
2840 IFPEEK(PO%+DS%)=160ORPEEK(PO%+DS%-22
)=135THEN2865 :rem 127
2845 POKEPO%-22,32:POKEPO%,32:PO%=PO%+DS%
:P%=PEEK(PO%):PP%=PEEK(PO%-22)
:rem 185
2850 POKEPO%+C%-22,1:POKEPO%+C%,1:POKEPO%
-22,153+DS%:POKEPO%,154+DS%:rem 164
2855 IFP%<>32ORPP%<>32THEN2900 :rem 246
2860 Q%=Q%+1:IFQ%<3THEN2840 :rem 0
2865 POKEPO%-22,32:POKEPO%,32:PO%=PO%+22:
P%=PEEK(PO%):PP%=PEEK(PO%-22)
:rem 99
2870 POKEPO%+C%-22,1:POKEPO%+C%,1:POKEPO%
-22,153+DS%:POKEPO%,154+DS%:rem 166
2875 IFP%<>32ORPP%<>32ORPEEK(PO%+22)=32TH
EN2900 :rem 4
2880 POKEV3,200:POKEV3,0:GOTO250:rem 160
2900 FORT=128TO255:POKEV2,T:POKEV3,383-T:
FORI=0TOTSTEP10:NEXT:NEXT:POKEV2,0:P
OKEV3,0 :rem 195
2905 POKEPO%,32:POKEPO%-22,32 :rem 163
2910 FORT=PO%TO7680STEP-22:P1%=PEEK(T-44)
:P2%=PEEK(T-22):P3%=PEEK(T):rem 43
2920 C1%=PEEK(T-44+C%):C2%=PEEK(T-22+C%):
C3%=PEEK(T+C%):POKET-44+C%,1:POKET-2
2+C%,1 :rem 131
2930 POKET+C%,1:POKET-44,158:POKET-22,159
:POKET,161:FORI=0TO250:NEXT:rem 129
2940 POKET-44,P1%:POKET-44+C%,C1%:POKET-2
2,P2%:POKET-22+C%,C2%:POKET,P3%:POKE
T+C%,C3% :rem 211
2950 NEXT:FORT=15TO1STEP-.15:POKEV,T:POKE
V1,244:POKEV2,244:POKEV3,244:NEXT
:rem 213
2960 FORT=15TO1STEP-.15:POKEV,T:POKEV1,24
1:POKEV2,241:POKEV3,241:NEXT
:rem 84
2970 POKEV1,0:POKEV2,0:POKEV3,0:POKEV,15:
GOTO270 :rem 220
3000 PRINT"{HOME}{WHT}IL TEMPO E' SCADUTO
!" :FORT=255TO128STEP-.1:POKEV2,T:P
OKEV3,T:NEXT :rem 22
3010 POKEV2,0:POKEV3,0:FORT=0TO1500:NEXT:
GOTO270 :rem 245
3100 SC=SC+FNA(1000):SC=SC+1000*L%:L%=L%+
1:GOTO90 :rem 250
3200 PO%=PO%-154:FORT=1TO12:PO%=PO%-1:POK
EPO%-22+C%,1:POKEPO%+C%,1:POKEPO%-22
,152 :rem 228
3210 POKEPO%,153:POKEV3,200:POKEV3,0:FORI
=0TO200:NEXT:POKEPO%,32:POKEPO%-22,3
2:NEXT :rem 135
3220 POKEPO%-22,152:POKEPO%,153:FORT=0TO4
6:POKEV3,N5(T)-1000*INT(N5(T)/1000)
:rem 162
3230 POKEV2,INT(N5(T)/1000):FORI=0TO250:N
EXT:NEXT:POKEV2,0:POKEV3,0:FORT=0TO2
500:NEXT :rem 98
3235 POKEPO%-44+C%,1 :rem 69
3240 FORT=1TO30:POKEPO%,32:POKEPO%-44,156
:POKEPO%-22,157:FORI=0TO100:NEXT
:rem 203
3250 POKEPO%-44,32:POKEPO%-22,156:POKEPO%
,157:POKEV3,200:POKEV3,0:FORI=0TO100
:NEXT:NEXT :rem 90
3260 T%=VAL(T$):T%=T%-100:IFT%<100THENT%=
100 :rem 217
3270 T$="000"+RIGHT$(STR$(T%),3):L%=1:SC=
SC+10000:GOTO90 :rem 109
3300 ONS%GOTO3400,3500,3600 :rem 196
3400 GOTO160 :rem 151
3500 DI%=DI%+I%:IFI%=1THENPOKEM%+DI%,32:P
OKEM%-DI%,32:GOTO3520 :rem 109
3510 POKEM%+DI%,160:POKEM%-DI%,160
:rem 135
3520 IFDI%=0THENPOKEM%,32:I%=1:GOTO3540
:rem 46
3530 IFDI%=5THENPOKEM%+5,160:POKEM%-5,160
:I%=-1 :rem 180
3540 GOTO160 :rem 156
3600 GOTO160 :rem 153
3700 ONS%GOTO3800,3900,4000 :rem 203
3800 IFM1%=8058THEND1%=22:POKEM1%,160:POK
EM1%+1,160:POKEM1%+2,160:GOTO3820
:rem 223
3810 IFM1%=8168THEND1%=-22 :rem 169
3820 IFM2%=8066THEND2%=22:POKEM2%,160:POK
EM2%+1,160:POKEM2%+2,160:GOTO3840
:rem 231
3830 IFM2%=8176THEND2%=-22 :rem 172
3840 M1%=M1%+D1%:M2%=M2%+D2%:IFD1%=-22THE
NPOKEM1%,32:POKEM1%+1,32:POKEM1%+2,3
2:GOTO3860 :rem 150
3850 POKEM1%+C%,6:POKEM1%+1+C%,6:POKEM1%+
2+C%,6:POKEM1%,160:POKEM1%+1,160:POK
EM1%+2,160 :rem 120
3860 IFD2%=-22THENPOKEM2%,32:POKEM2%+1,32
:POKEM2%+2,32:GOTO3880 :rem 151
3870 POKEM2%+C%,6:POKEM2%+1+C%,6:POKEM2%+
2+C%,6:POKEM2%,160:POKEM2%+1,160:POK
EM2%+2,160 :rem 128
3880 IFPO%-22=M1%ORPO%-22=M2%ORPO%-22=M1%
+1ORPO%-22=M2%+1ORPO%-22=M1%+2THEN29
00 :rem 0
3885 IFPO%-22=M2%+2THEN2900 :rem 244
3890 GOTO160 :rem 164
3900 IFZ%=1THEN3920 :rem 69
3910 LA%=LA%(FNA(7)):Z%=1 :rem 67
3920 POKELA%,32:LA%=LA%+44:POKELA%+C%,7:P
OKELA%,142:IFLA%=PO%-22THEN2900
:rem 164
3930 IFPEEK(LA%+44)=160THENPOKELA%,32:POK
ELA%-88,142:Z%=0 :rem 47
3940 GOTO160 :rem 160
4000 DI%=DI%+I%:IFI%=1THEN4010 :rem 151
4005 POKEU1%+DI%*22,32:POKEW1%-DI%*22,32:
POKEU2%+DI%*22,32:POKEW2%-DI%*22,32:
GOTO4020 :rem 18
4010 POKEU1%+C%+DI%*22,6:POKEW1%+C%-DI%*2
2,6:POKEU1%+DI%*22,160:POKEW1%-DI%*2
2,160 :rem 255

```



```

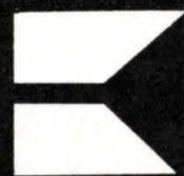
4015 POKEU2%+C%+DI%*22,6:POKEW2%+C%-DI%*2
      2,6:POKEU2%+DI%*22,160:POKEW2%-DI%*2
      2,160 :rem 8
4020 IFDI%=3THENI%=-1:POKEU1%+66,32:POKEU
      2%+66,32:POKEW1%-66,32:POKEW2%-66,32
      :GOTO4030 :rem 227
4025 IFDI%=0THENI%=1:POKEU1%,160:POKEU2%,
      160:POKEW1%,160:POKEW2%,160:rem 230
4030 GOTO160 :rem 151
4100 ONS%GOTO4200,4300,4400 :rem 192
4200 GOTO160 :rem 150
4300 POKEM2%,32:IFPO%=M2%-22THENPOKEPO%-2
      2,32:POKEPO%,32:PO%=PO%+I2%:rem 246
4310 M2%=M2%+I2%:POKEM2%+C%,3:POKEM2%,160
      :rem 231
4320 IFPO%=M2%-22THENPOKEPO%+C%,1:POKEPO%
      -22+C%,1:POKEPO%-22,153+DS%:POKEPO%,
      154+DS% :rem 146
4330 IFM2%=8016THENI2%=1:GOTO4350
      :rem 133
4340 IFM2%=8026THENI2%=-1 :rem 117
4350 GOTO160 :rem 156
4400 M1%=M1%+I1%:IFI1%=-1THENPOKEM1%+C%,6
      :POKEM1%,160:GOTO4420 :rem 26
4410 POKEM1%,32 :rem 44
4420 IFM1%=7862THENI1%=1:POKEM1%,32:GOTO4
      440 :rem 40
4430 IFM1%=7872THENI1%=-1:POKEM1%,160
      :rem 74
4440 GOTO160 :rem 156
4500 ONS%GOTO4600,4700,4800 :rem 208
4600 Z%=0 :rem 182
4610 POKEB1%,32:B1%=B1%+I1%:POKEB1%+C%,4:
      POKEB1%,81:IFB1%=PO%THEN2900
      :rem 122
4620 IFB1%=8143THENI1%=1:GOTO4640
      :rem 125
4630 IFB1%=8162THENI1%=-1 :rem 107
4640 Z%=Z%+1:IFZ%<2THEN4610 :rem 21

```

```

4650 GOTO160 :rem 159
4700 POKEB2%,32:B2%=B2%+I2%:POKEB2%+C%,5:
      POKEB2%,81:IFB2%=PO%THEN2900
      :rem 130
4710 POKEB3%,32:B3%=B3%+I3%:POKEB3%+C%,2:
      POKEB3%,81:IFB3%=PO%THEN2900
      :rem 135
4720 IFB2%=7989THENI2%=1:GOTO4740
      :rem 146
4730 IFB2%=8008THENI2%=-1 :rem 109
4740 IFB3%=7989THENI3%=1:GOTO4760
      :rem 152
4750 IFB3%=8008THENI3%=-1 :rem 113
4760 GOTO160 :rem 161
4800 POKEB4%,32:B4%=B4%+I4%:POKEB4%+C%,FN
      A(2):POKEB4%,81:IFB4%=PO%THEN2900
      :rem 180
4810 IFB4%=7835THENI4%=1:GOTO4830
      :rem 140
4820 IFB4%=7854THENI4%=-1 :rem 121
4830 GOTO160 :rem 159
4900 ONS%GOTO5000,5100,5200 :rem 197
5000 IFZ1%=0THENP1%=M1%:Z1%=1:GOTO5020
      :rem 184
5010 POKEP1%,32 :rem 44
5020 P1%=P1%-1:POKEP1%+C%,7:POKEP1%,140:I
      FP1%=PO%THEN2900 :rem 237
5030 IFPEEK(P1%-1)=160ORRND(1)<.1THENZ1%=
      0:POKEP1%,32 :rem 51
5040 GOTO160 :rem 153
5100 FORT=0TO4:POKEFA%(T),32:FA%(T)=FA%(T
      )+DF%(T):POKEFA%(T)+C%,FNA(8):POKEFA
      %(T),141 :rem 123
5110 IFFA%(T)=PO%ORFA%(T)=PO%-22THEN2900
      :rem 38
5120 IFPEEK(FA%(T)+DF%(T))=160THENDF%(T)=
      -DF%(T) :rem 178
5130 NEXT:GOTO160 :rem 18
5200 IFZ2%=0THENP2%=M2%:Z2%=1:GOTO5220
      :rem 192
5210 POKEP2%,32 :rem 47
5220 P2%=P2%+1:POKEP2%+C%,7:POKEP2%,140:I
      FP2%=PO%THEN2900 :rem 242
5230 IFPEEK(P2%+1)=137ORRND(1)<.1THENZ2%=
      0:POKEP2%,32 :rem 58
5240 IFZ3%=0THENP3%=M3%:Z3%=1:GOTO5260
      :rem 204
5250 POKEP3%,32 :rem 52
5260 P3%=P3%-1:POKEP3%+C%,7:POKEP3%,140:I
      FP3%=PO%THEN2900 :rem 253
5270 IFPEEK(P3%-1)=139ORRND(1)<.1THENZ3%=
      0:POKEP3%,32 :rem 69
5280 GOTO160 :rem 159
5300 POKEM%,32:IFM%-22=PO%THENPOKEPO%-22,
      32:POKEPO%,32:PO%=PO%+I1% :rem 146
5310 M%=M%+I1%:POKEM%+C%,6:POKEM%,160:IFP
      O%<>M%-22THEN5330 :rem 38
5320 POKEPO%-22+C%,1:POKEPO%+C%,1:POKEPO%
      -22,153+DS%:POKEPO%,154+DS%:rem 159
5330 IFM%=8167THENI1%=1:GOTO5350:rem 91
5340 IFM%=8171THENI1%=-1 :rem 68
5350 IFZ%=0THENZ%=1:LA%=LA%(FNA(3))
      :rem 233
5360 POKELA%,32:LA%=LA%+44:POKELA%+C%,7:P
      OKELA%,142:IFLA%=PO%-22THEN2900
      :rem 164
5370 IFPEEK(LA%+44)=160THENZ%=0:POKELA%,3
      2:POKELA%-88,142 :rem 47
5380 POKEB%,32:B%=B%+I2%:POKEB%+C%,FNA(2)

```

Scritte giganti

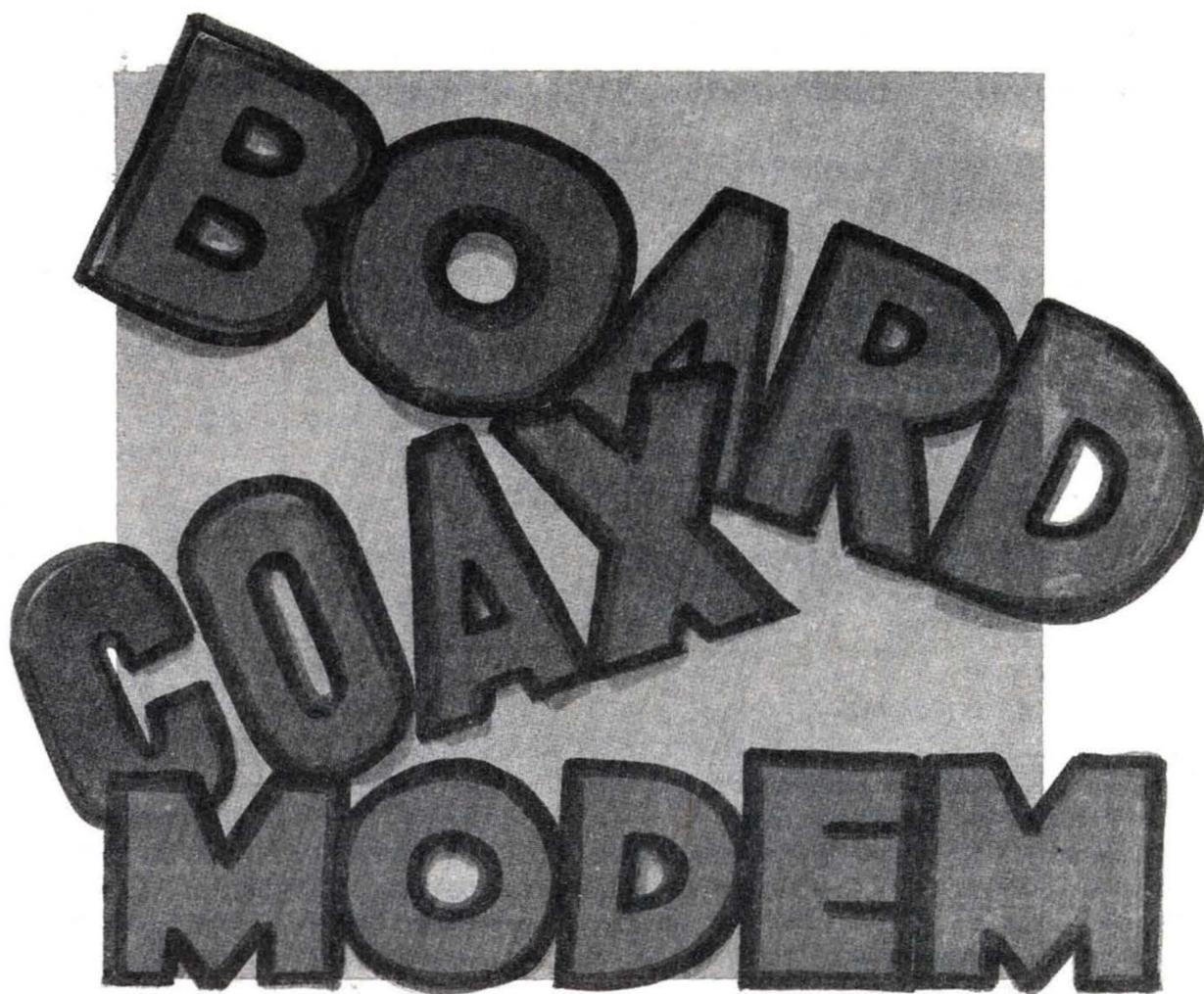
di **D. Smoak**
trad. e adatt.
di **F. Sarcina**

Con questo programma, usabile con tutte le stampanti a matrice di punti Commodore, potrete costruirvi le vostre personalizzate scritte giganti.

Utilizzate il vostro computer e la vostra stampante in maniera creativa con questo breve programma BASIC. SCRITTE GIGANTI è facile da usare; vi permette di selezionare la larghezza e l'altezza dei caratteri. Inoltre vi è la possibilità di utilizzare un set ridefinito di simboli, in cui potrete includere lettere speciali o qualsiasi cosa vi passa per la testa.

Uso dei menu

Digitate l'opportuna versione del programma per il vostro computer. Dopo averne salvata una copia date il RUN e assicuratevi che il numero di periferica della vostra stampante sia il 4. A questo punto il programma vi domanda se volete usare il set di caratteri maiuscolo-grafico, il maiuscolo-minuscolo o un set personale da voi precedentemente definito. Dopo aver effettuato la scelta vi si chiederà con quale carattere volete che le lettere giganti vengano costruite. Ora è il momento di inserire il messaggio che volete stampare; esso può avere la lunghezza di due righe di schermo. Dovete quindi impostare l'altezza delle vostre lettere, un numero fra 1 e 9; scegliendo 1 otterrete delle lettere alte 7 caratteri normali, con 2 le lettere saranno alte 14 caratteri e via di seguito fino a 9, a cui corrispondono 63 caratteri di altezza. Il successivo parametro è la larghezza, che deve essere un numero compreso tra 1 e il precedente valore impostato per l'altezza. A questo punto inizia la stampa. Ricordate che, ovviamente, la velocità di stampa risulta essere proporzionale alla grandezza dei caratteri stampati: regolatevi quindi di conseguenza, se dovete stampare molti messaggi.



Modifiche al programma per il funzionamento con 1526 ed MPS-802

Questo programma funzionerà senza il bisogno di alcuna modifica sulle stampanti 1525, MPS-801 e MPS-803. Invece su 1526 ed MPS-802 occorrerà fare i seguenti piccoli cambiamenti. Per prima cosa aggiungere all'inizio del programma la seguente istruzione:

```
1 OPEN6,4,6: PRINT # 6, CHR$(27)
: CLOSE6
```

Ciò serve per modificare lo spazio fra

una riga di stampa e la successiva, diminuendolo di un po'. Se volete i caratteri ancora più ravvicinati, usate un numero inferiore a 27 nella funzione CHR\$; viceversa per spazi maggiori.

L'ultima modifica è cambiare i CHR\$(8) in CHR\$(32) alle linee 360 e 390 (o alle linee 340 e 370 nella versione per VIC20).

Uso dei caratteri ridefiniti

Per personalizzare ulteriormente le vostre scritte potete usare, come già detto, un set di caratteri da voi precedente-

mente ridefinito. Se siete un programmatore sufficientemente esperto, la questione dovrebbe essere semplice per voi. Il metodo di ridefinizione viene comunque spiegato abbastanza esaurientemente sulla Guida di Riferimento

del vostro computer. Ricordatevi comunque che il nuovo set di caratteri dovrà essere posizionato a partire da 13312 per i possessori di Plus/4 e C16, mentre nel VIC può essere messo in qualunque zona di memoria che sicura-

mente non viene "sporcata" dal programma o dalle sue variabili. In ambedue le versioni, accertatevi di aver caricato in memoria il nuovo set prima di caricare e mandare in esecuzione SCRITTE GIGANTI.

Scritte giganti per VIC 20

```

100 POKE56,28:POKE55,0:CLR          :rem 11
110 CC=0:PRINT"{CLR}{BLK}{RVS}{ 3 SPAZI}
    SCRITTE GIGANTI{ 4 SPAZI}{OFF}":PRINT
    "{ 2 GIU'}PREMI UN TASTO:" :rem 138
120 PRINT"{ 2 GIU'}{RVS}U{OFF}+PER SET MA
    IUSCOLO/{ 4 SPAZI}GRAFICO":PRINT"
    { 2 GIU'}{RVS}L{OFF}+PER MAIUSCOLO/MI
    NU-{ 3 SPAZI}SCOLO" :rem 171
130 PRINT"{ 2 GIU'}{RVS}C{OFF}+PER CARATT
    ERI PER-{ 4 SPAZI}SONALIZZATI
    { 11 SPAZI}(RICORDATI I DATI!)"
    :rem 61
140 GETCH$:IFCH$=""THEN140          :rem 225
150 IFCH$="U"THENCH=32768:PRINTCHR$(142):
    GOTO190 :rem 104
160 IFCH$="L"THENCH=32768+2048:PRINTCHR$(
    14):GOTO190 :rem 39
170 IFCH$="C"THENFORK=7424TO7431:POKEK,0:
    NEXT:POKE36869,255:CH=7168:GOTO190
    :rem 42
180 GOTO140 :rem 103
190 PRINT"{CLR}{RVS}CARATTERE DI RIEMPI-
    { 2 SPAZI}MENTO":GOSUB490 :rem 5
200 PP$=MT$:PRINT"{CLR}{RVS}QUAL E' IL ME
    SSAGGIO":GOSUB 490:N$=MT$ :rem 224
210 PRINT"{RVS}QUANTO ALTO{ 11 SPAZI}(DA
    1 A 9)":INPUTTS:IFTS<1THENTS=1
    :rem 145
220 PRINT"{RVS}QUANTO LARGO{ 9 SPAZI}(DA
    1 A"TS"{SIN})":INPUTWS:IFWS>TSORWS<1T
    HENWS=1 :rem 95
230 OPEN1,4 :rem 88
240 FORL=1TOLEN(N$) :rem 124
250 A$=MID$(N$,L,1):PRINT"{RVS}{HOME}
    { 10 GIU'}STAMPA{OFF} "A$:IFA$=" "THE
    NGOSUB460:GOTO360 :rem 106
260 Q=ASC(A$):Q=Q+(Q>63)*64+(Q>191)*64+(Q
    =255)*33 :rem 68
270 I=CH+Q*8 :rem 136
280 FORJ=7TO0STEP-1 :rem 172
290 FORK=7TO0STEP-1 :rem 174
300 B=PEEK(I+K) :rem 71
310 IFBAND2↑JTHENFORT=1TOTS:PR$=PR$+PP$:N
    EXT:GOTO330 :rem 252
320 FORT=1TOTS:PR$=PR$+" ":NEXTT:rem 199
330 NEXTK :rem 32
340 FORX=1TOWS:PRINT#1,CHR$(15)SPC((72-TS
    *8)/2)PR$CHR$(8):NEXTX:PR$=""
    :rem 186
350 NEXTJ :rem 33
360 NEXTL :rem 36
370 PRINT#1,CHR$(8):CLOSE1 :rem 18
380 POKE36879,26:IFPEEK(36869)=255THENCC=
    1:POKE36869,240 :rem 245
390 PRINT"{CLR}PREMI UN TASTO:" :PRINT"
    { 2 GIU'}X PER FINIRE":PRINT"
    { 2 GIU'}R PER RISTAMPARE" :rem 108
400 PRINT"{ 2 GIU'}N PER NUOVO MESSAGGIO"
    :rem 242

```

```

410 GETY$:IFY$=""THEN410           :rem 125
420 IFY$="X"THENOPEN1,4:PRINT#1,CHR$(15):
    CLOSE1:POKE36879,27:PRINT"{CLR}":END
    :rem 11
430 IFY$="R"THENGOSUB470:PRINT"{CLR}"N$:G
    OTO230 :rem 216
440 IFY$="N"THEN110 :rem 52
450 GOTO410 :rem 103
460 FORX=1TO8*WS:PRINT#1,CHR$(8):NEXT:RET
    URN :rem 90
470 IFCC=1THENPOKE36869,255:CC=0:rem 129
480 RETURN :rem 124
490 POKE198,5:POKE631,5:POKE632,34:POKE63
    3,34:POKE634,20:POKE635,144 :rem 194
500 MT$="":INPUTMT$:RETURN :rem 132

```

Scritte giganti per C16-Plus/4

```

100 POKE56,52:POKE55,0:CLR:CU=65298:GOSUB
    520:SYS819:COLOR0,2,5 :rem 249
110 PRINT"{CLR}{BLK}{RVS}{ 12 SPAZI}SCRIT
    TE GIGANTI{ 13 SPAZI}{OFF}":rem 75
120 CC=0:PRINT"{ 3 GIU'}PREMI UN TASTO:"
    PRINT"{ 2 GIU'}{RVS}U{OFF}+PER SET MA
    IUSCOLO/GRAFICO" :rem 215
130 PRINT"{ 2 GIU'}{RVS}L{OFF}+PER MAIUSC
    OLO/MINUSCOLO" :rem 81
140 PRINT"{ 2 GIU'}{RVS}C{OFF}+PER CARATT
    ERI PERSONALIZZATI (I DATI{ 2 SPAZI}
    DEVONO ESSERE MESSI DA 13312!)"
    :rem 2
150 GETCH$:IFCH$=""THEN150          :rem 227
160 IFCH$="U"THENCH=14336:PRINTCHR$(142):
    GOTO200 :rem 88
170 IFCH$="L"THENCH=14336+1024:PRINTCHR$(
    14):GOTO200 :rem 16
180 IFCH$="C"THENFORK=13568TO13575:POKEK,
    0:NEXT:CH=13312:GOSUB510:GOTO200
    :rem 194
190 GOTO150 :rem 105
200 PRINT"{CLR}{RVS}CARATTERE DI RIEMPIE
    MENTO" :rem 121
210 GOSUB590:PP$=MT$ :rem 177
220 PRINT"{CLR}{RVS}QUAL E' IL TUO MESSAG
    GIO{ 2 SPAZI}":GOSUB590:N$=MT$
    :rem 219
230 PRINT"{RVS}QUANTO ALTO (DA 1 A 9)":IN
    PUTTS:IFTS<1THENTS=1 :rem 147
240 PRINT"{RVS}QUANTO LARGO (DA 1 A"TS"
    {SIN})":INPUTWS:IFWS>TSORWS<1THENWS=T
    S :rem 215
250 OPEN1,4 :rem 90
260 FORL=1TOLEN(N$) :rem 126
270 A$=MID$(N$,L,1):PRINT"{RVS}{HOME}
    { 9 GIU'}STAMPA{OFF} "A$:IFA$=" "THEN
    GOSUB480:GOTO380 :rem 95
280 Q=ASC(A$):Q=Q+(Q>63)*64+(Q>191)*64+(Q
    =255)*33 :rem 70
290 I=CH+Q*8 :rem 138
300 FORJ=7TO0STEP-1 :rem 165

```

```

310 FORK=7TO0STEP+1           :rem 167
320 B=PEEK(I+K)                :rem 73
330 IFBAND2↑JTHENFORT=1TOTS:PR$=PR$+PP$:N
    EXTT:GOTO350                :rem 0
340 FORT=1TOTS:PR$=PR$+" ":NEXTT:rem 201
350 NEXTK                       :rem 34
360 FORX=1TOWS:PRINT#1,CHR$(15)SPC((72-TS
    *8)/2)PR$CHR$(8):NEXTX:PR$=""
                                :rem 188
370 NEXTJ                       :rem 35
380 NEXTL                       :rem 38
390 PRINT#1,CHR$(8):CLOSE1      :rem 20
400 IFCH$="C"THENPOKECU,196:POKECU+1,208:
    CC=1                         :rem 169
410 PRINT"{CLR}PREMI UN TASTO:"PRINT"
    {GIU'}X PER FINIRE":PRINT"{GIU'}R PER
    RISTAMPARE IL MESSAGGIO"     :rem 119
420 PRINT"{GIU'}N PER UN NUOVO MESSAGGIO"
                                :rem 134
430 GETY$:IFY$=""THEN430       :rem 129
440 IFY$="R"THENGOSUB490:PRINT"{CLR}"N$:G
    OTO250                        :rem 221
450 IFY$="X"THENOPEN1,4:PRINT#1,CHR$(15):
    CLOSE1:PRINT"{CLR}":END      :rem 255
460 IFY$="N"THEN110            :rem 54
470 GOTO430                    :rem 107
480 FORX=1TO8*WS:PRINT#1,CHR$(8):NEXT:RET
    URN                          :rem 92
490 IFCC=1THENCC=0:GOSUB510    :rem 146
500 RETURN                     :rem 117
510 POKECU,PEEK(CU)AND251:POKECU+1,PEEK(C
    U+1)AND3OR52:RETURN          :rem 29
520 FORI=819TO850:READA:POKEI,A:X=X+A:NEX
    T                             :rem 168
530 IFX<>5848THENPRINT"ERRORE NELLE ISTRU
    ZIONI DATA":STOP           :rem 203
540 RETURN                     :rem 121
550 DATA169,56,133,254,169,208,133,252,16
    9,0                          :rem 183
560 DATA133,251,133,253,168,162,8,177,251
                                :rem 28
570 DATA145,253,200,208,249,230,252,230,2
    54                          :rem 115
580 DATA202,208,242,96        :rem 112
590 POKE239,5:POKE1319,155:POKE1320,34:PO
    KE1321,34:POKE1322,20:POKE1323,144
                                :rem 5
600 INPUTMT$:RETURN           :rem 5

```

**BOWARD
COAXARD
MODEM**

**GUIDA AL COMMODORE
PLUS 4
LAVORIAMO CON IL
COMMODORE 16**

LIBRI

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano - Tel. 6880951/2/3/4/5

cod. CC256 L. 30.000

cod. CC244 L. 20.000

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

Cod.Cod.Cod.C.A.P.Prov.

Cod.Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul cc. n° 11666203 intestato a Gruppo Editoriale Jackson
- Milano

Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i
soci in Italia)

Nome

Cognome

Via

Città

Se richiesta fattura:
Cod. F. e P. Iva

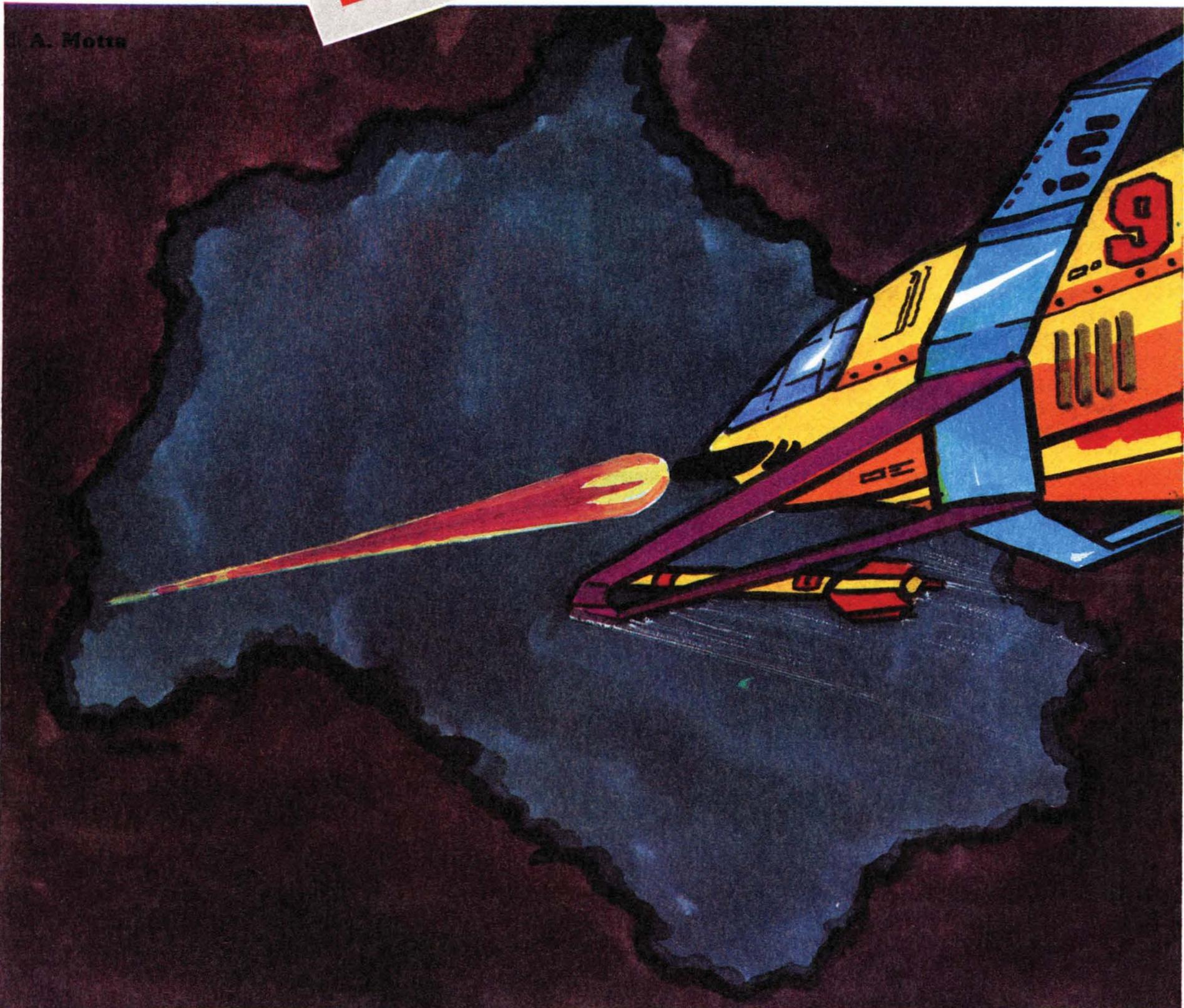
Data

Firma

Per i residenti all'estero: pagamento anticipato (vaglia o versamento su ns. ccp)



Asteroidi per C16-Plus/4



Il gioco consiste nel guidare la vostra astronave attraverso una zona piena di asteroidi e di sopravvivere il più a lungo possibile.

Avete a disposizione tre astronavi e per ciascuna di esse cinque scudi di forza magnetica, che vi permetteranno di disintegrare gli asteroidi che non riuscirete a evitare. Dopo cinque collisioni l'astronave sarà distrutta. I coman-

di sono: destra e sinistra utilizzano i relativi cursori; barra spaziatrice per attivare lo scudo magnetico. Quando tutte le vostre astronavi saranno distrutte il gioco avrà termine e vi sarà mostrata la durata della vostra missione e il tempo record sino a quel momento raggiunto. Notevole la ridefinizione dei caratteri (30), dei quali ben 13 per gli asteroidi, 4 per l'astronave, 7 per lo scoppio e 6 per le stelle del firmamento.

Da segnalare l'utilizzo del tasto ESCAPE (CHR\$(27)), che ha permesso la definizione di una finestra a video, in modo da effettuare lo scroll verso l'alto senza intaccare la prima linea dove sono mostrate le astronavi rimaste e gli scudi a disposizione inutilizzati.

- Variabili utilizzate:**
- NA numero delle astronavi rimaste
 - CF scudi magnetici da utilizzare

X	posizione dell'asse verticale a video della astronave	CO	sulla 25ª riga numero delle collisioni	N	cord valore del vostro tempo
X1	posizione verticale precedente della astronave	N\$	stringa contenente il tempo della missione	HI	valore del tempo record
X2 e X3	punti di stampa degli asteroidi	HI\$	stringa contenente il tempo re-	SR\$	stringa contenente le stelle

Asteroidi per C16-Plus/4

```

1010 POKE56,60:POKE52,60:POKE55,0:POKE51,0 :rem 170
1020 GOSUB9000:GOSUB7000 :rem 141
1100 COLOR0,7,0:COLOR4,7,0 :rem 220
1110 PRINT"{CLR} {CYN}ASTRONAVI {YEL}
[<F>] [<F>] [<F>]":CHAR,24,0,"{CYN}S
CUDI" :rem 25
1120 PRINT"{HOME}{ 2 GIU' }"CHR$(27)"T":PR
INTCHR$(27)"L" :rem 47
1130 N$="000000":NA=3 :rem 17
1140 CF=5:CO=0:X=18:X1=X :rem 138
1150 FORI=1TOCF:POKE3100+2*I,90 :rem 35
1160 POKE2076+2*I,37:NEXT :rem 165
1170 FORI=1TO22:PRINTMID$(SR$,INT(RND(0)*
29)+1,40);:NEXT :rem 79
1180 CHAR,X,8,"{YEL}[<U>][<O>]@" :rem 9
1190 FORI=1TO1000:NEXT:TI$=N$ :rem 240
1195 POKE65297,21:POKE65294,X*3:POKE65298
,1 :rem 195
1200 GETA$:X=X+(A$="{SIN}"ANDX>0)-(A$="{
DES}"ANDX<36) :rem 24
1210 IFA$=" "THEN1500 :rem 46
1220 IFPEEK(3432+X)>99ORPEEK(3433+X)>99OR
PEEK(3434+X)>99THEN1600 :rem 238
1230 GOSUB1400 :rem 11
1240 CHAR,X1,7,"B D":CHAR,X,8,"{YEL}[<U>]
[<O>]@" :X1=X :rem 196
1250 POKE65294,X*3 :rem 226
1260 GOTO1200 :rem 197
1400 X2=INT(RND(0)*40):X3=INT(RND(TI)*40)
:rem 89
1410 POKE3992+X2,INT(RND(0)*13)+100:POKE2
968+X2,56 :rem 38
1420 POKE3992+X3,INT(RND(0)*13)+100:POKE2
968+X3,56 :rem 41
1430 CHAR,0,24,MID$(SR$,INT(RND(0)*29)+1,
39) :rem 110
1440 PRINT"{HOME}{ 2 GIU' }"CHR$(27)"T"CHR
$(27)"V" :rem 119
1460 RETURN :rem 171
1500 IFCF=0THEN1220:VOL8 :rem 162
1510 POKE3100+2*CF,32:CF=CF-1 :rem 66
1520 GOSUB1400 :rem 13
1530 CHAR,X1,7," E " :rem 52
1540 CHAR,X,8,"{WHT}[<L>][<W>][<Y>]":SOUN
D3,1000,30 :rem 215
1550 FORI=1TO200:NEXT :rem 23
1560 CHAR,X,8,"{YEL}[<U>][<O>]@" :X1=X:GOT
O1195 :rem 166
1600 CO=CO+1:IFCO=5THEN1700 :rem 72
1610 GOSUB1400:CHAR,X1,7,"G D":CHAR,X,8,"
{RED}[<U>][<O>]@" :rem 111
1620 VOL8:SOUND3,1000,5:COLOR0,2:rem 125
1630 SOUND3,800,5:COLOR0,7,0 :rem 83
1640 CHAR,X,8,"{YEL}[<U>][<O>]@" :X1=X:GOT
O1195 :rem 165
1700 N$=TI$:GOSUB1400:POKE65297,PEEK(6529
7)AND15 :rem 21
1710 CHAR,X1,7," E " :CHAR,X,8,"{WHT}
{FLASH ON}[<E>][<E>] {WHT} :rem 131
1720 FORI=5TO8:VOLI :rem 187
1730 SOUND3,550,7:NEXT :rem 41
1740 CHAR,X,7,"{WHT}[<R>][<W>][<H>]{GIU' }
{ 3 SIN}[<J>] [<J>]{GIU' }{ 3 SIN}
[<L>][<W>][<Y>]" :rem 185
1750 FORI=8TO1STEP-1:VOLI :rem 84
1760 SOUND3,700,10:NEXT :rem 83
1770 POKE3081+2*NA,32:NA=NA-1 :rem 100
1780 CHAR,X,7,"EG {GIU' }{ 3 SIN} D {GIU' }
{ 3 SIN}BDG" :rem 178
1790 FORI=1TO500:NEXT :rem 32
1800 CHAR,5,12,"{YEL}L'ASTRONAVE E' STATA
DISTRUTTA":VOL8 :rem 27
1810 FORI=1TO1000:NEXT:IFNA=0THEN1900
:rem 3
1820 PRINT"{HOME}{ 2 GIU' }"CHR$(27)"T
{CLR}":GOTO1140 :rem 240
1900 T$=LEFT$(N$,2)+". "+MID$(N$,3,2)+". "+
RIGHT$(N$,2) :rem 192
1910 N=VAL(N$):PRINT"{CLR}" :rem 154
1920 CHAR,2,5,"{YEL}LA TUA MISSIONE E' DU
RATA: ":PRINTT$ :rem 59
1930 IFN>HITHE2000 :rem 106
1940 CHAR,2,10,"{CYN}IL RECORD DI DURATA
E': ":PRINTHI$ :rem 170
1950 CHAR,2,20,"{WHT}GIOCHI ANCORA? (S/N)
" :rem 21
1960 GETA$:IFA$="N"THENPRINT"{CLR}":END
:rem 182
1970 IFA$<>"S"THEN1960 :rem 213
1980 GOTO1100 :rem 205
2000 HI$=T$:HI=N :rem 130
2010 CHAR,2,8,"HAI FATTO IL NUOVO RECORD
DI DURATA" :rem 156
2020 FORI=500TO700STEP4 :rem 112
2030 SOUND1,1,1:NEXT :rem 202
2040 GOTO1950 :rem 206
7000 REM ISTRUZIONI :rem 203
7010 COLOR0,11,1:COLOR4,11,1 :rem 58
7020 PRINT"{CLR}":CHAR,11,8,"{YEL}A S T E
R O I D I" :rem 41
7030 CHAR,7,11,"{WHT}VUOI LE ISTRUZIONI?
(S/N)" :rem 162
7040 GETA$:IFA$="N"THENRETURN :rem 28
7050 IFA$<>"S"THEN7040 :rem 203
7060 PRINT"{CLR}":CHAR,11,0,"{YEL}A S T E
R O I D I" :rem 37
7070 PRINT:PRINT"{ 2 GIU' }{WHT}LA TUA NAV
E SPAZIALE E' ENTRATA IN{ 2 SPAZI}UN
A" :rem 125
7080 PRINT"ZONA DI ASTEROIDI: QUANTO TEMP
O RIUSCI-" :rem 156
7090 PRINT"RAI A SOPRAVVIVERE PRIMA CHE V
ENGA{ 2 SPAZI}DI-" :rem 222
7100 PRINT"STRUTTA?" :rem 15
7110 PRINT"PER MUOVERTI USA I CURSORI DES
TRO E SI-" :rem 52
7120 PRINT"NISTRO.{ 2 SPAZI}HAI A DISPOSI
ZIONE UNO SCHERMO" :rem 94
7130 PRINT"AD ENERGIA CHE TI PERMETTERA'
DI DISIN-" :rem 218

```

LOAD



ASTEROIDI
PER C16

```
7140 PRINT"TEGRARE GLI ASTEROIDI CHE NON
      RIUSCIRAI" :rem 141
7150 PRINT"AD EVITARE. ATTENZIONE PERO':
      LO SCHER-" :rem 54
7160 PRINT"MO'E' DISPONIBILE PER SOLI CIN
      QUE SCON-" :rem 28
7170 PRINT"TRI E PER ATTIVARLO PREMI LO S
      PAZIO." :rem 141
7180 PRINT"DOPO CINQUE COLLISIONI LA NAVE
      ESPLODE-" :rem 125
7190 PRINT"RA'." :rem 138
7200 PRINT"{ 2 GIU'}{CYN}PREMI UN TASTO P
      ER INIZIARE" :rem 72
7210 GETA$:IFA$=""THEN7210 :rem 183
7220 RETURN :rem 171
8000 REM DATA L/M :rem 142
8010 DATA160,0,185,0,208,153,0,60
      :rem 117
8020 DATA185,0,209,153,0,61,185,0
      :rem 127
8030 DATA210,153,0,62,185,0,211,153
      :rem 216
8040 DATA0,63,24,136,208,228,96 :rem 44
8100 REM DATA NUOVI CARATTERI :rem 247
8110 DATA62,63,255,255,255,127,63,26
      :rem 40
8120 DATA112,244,254,127,255,255,126,32
      :rem 178
8130 DATA0,0,24,60,60,56,0,0 :rem 119
8140 DATA0,0,48,56,112,32,0,0 :rem 171
8150 DATA0,24,60,56,14,4,0,0 :rem 124
8160 DATA112,248,248,124,56,48,0,0
      :rem 187
8170 DATA0,16,56,56,60,16,0,0 :rem 184
8180 DATA0,0,30,30,62,84,23,0 :rem 174
8190 DATA94,255,252,252,62,127,60,60
      :rem 41
8200 DATA0,62,127,126,62,31,14,0:rem 68
8210 DATA96,240,240,122,14,63,127,126
      :rem 74
8220 DATA62,127,254,15,247,248,240,96
      :rem 93
8230 DATA24,60,14,7,6,4,0,0 :rem 77
8240 DATA145,82,16,7,244,8,74,137
      :rem 150
8250 DATA0,4,34,18,1,68,50,9 :rem 142
8260 DATA0,8,170,42,73,65,137,137
      :rem 143
8270 DATA0,32,68,72,128,36,76,144
      :rem 149
8280 DATA100,24,198,0,222,0,24,102
      :rem 168
8290 DATA9,50,68,1,18,34,4,0 :rem 146
8300 DATA144,76,34,128,72,68,32,0
      :rem 141
8310 DATA0,0,3,14,57,255,6,24 :rem 182
```

```
8320 DATA24,126,231,126,196,255,0,0
      :rem 229
8330 DATA0,0,192,112,156,255,96,24
      :rem 185
8340 DATA0,0,24,60,102,255,36,0 :rem 17
8350 DATA0,0,0,0,0,0,0,8 :rem 166
8360 DATA128,0,0,0,0,0,0,0 :rem 10
8370 DATA0,0,0,0,32,0,0,0 :rem 213
8380 DATA0,4,0,0,0,0,64,0 :rem 223
8390 DATA0,0,0,64,0,0,0,0 :rem 220
8400 DATA0,128,0,0,0,0,2,0 :rem 7
9000 REM TRASFERIMENTO CARATTERI IN RAM
      :rem 166
9010 PRINT"{CLR}":CHAR,13,11,"A T T E N D
      I" :rem 29
9020 T=0:FORI=1000TO1030:READA :rem 242
9030 POKEI,A:T=T+A:NEXT :rem 181
9040 IFT<>3288THENPRINT"{CLR}ERRORE NEI D
      ATA!!!":END :rem 203
9050 SYS1000:VOL8 :rem 241
9060 POKE65298,PEEK(65298)AND251:rem 39
9070 POKE65299,PEEK(65299)AND3OR60
      :rem 204
9100 REM MODIFICA CARATTERI :rem 137
9110 FORI=16160TO16351:READA :rem 112
9120 POKEI,A:NEXT :rem 42
9130 FORI=15888TO15935:READA :rem 137
9140 POKEI,A:NEXT :rem 44
9500 REM VARIABILI INIZIALI :rem 158
9510 SR$="{ 2 SPAZI}C{SPAZI}F{ 2 SPAZI}G
      { 2 SPAZI}B{SPAZI}E{SPAZI}D{SPAZI}B
      { 2 SPAZI}EF{ 3 SPAZI}G{ 2 SPAZI}C
      { 3 SPAZI}B {SPAZI}E{SPAZI}C
      { 2 SPAZI}FD{ 2 SPAZI}E{SPAZI}F
      { 2 SPAZI}B D{ 3 SPAZI}FC{ 2 SPAZI}
      G{ 2 SPAZI}B{ 2 SPAZI}E D C "
      :rem 174
9530 HI$="00.00.00":HI=0 :rem 187
9900 RETURN :rem 178
```



Seqfiling System
1.0 per VIC 20

di M. Cristuib Grizzi

Dopo la pubblicazione sul numero di Ottobre dello scorso anno di "Relfiling System 1.0", il database per sistema a dischi con utilizzo dei file relativi, ci sono pervenute molte richieste da parte dei lettori per un database con le stesse caratteristiche dedicato invece esclusivamente al registratore a cassette.

Con questo programma abbiamo cercato di offrire un valido strumento di archivio anche a coloro che non posseggono un drive, sperando di essere riusciti a realizzare un database per nastro magnetico ugualmente veloce e flessibile. Il programma richiede un' espansione di memoria da almeno 8 Kbyte.

La filosofia del programma

La limitazione posta dall'utilizzo forzato del registratore a cassette impone infatti la programmazione del software per archivio nella forma sequenziale: il termine "sequenziale", già di per sé esemplificativo, indica che i dati vengono letti o scritti in sequenza: per vedere che cosa contiene il centesimo record siamo obbligati a leggere anche tutti i precedenti novantanove, anche se il loro contenuto non ci interessa.

Inoltre, se disponiamo di un file con cento record e desideriamo aggiungervi il centounesimo, non sarà possibile - come avviene per i file relativi su disco - "incollare" questo record alla fine del file, ma dovremo riscrivere per intero tutto il file.

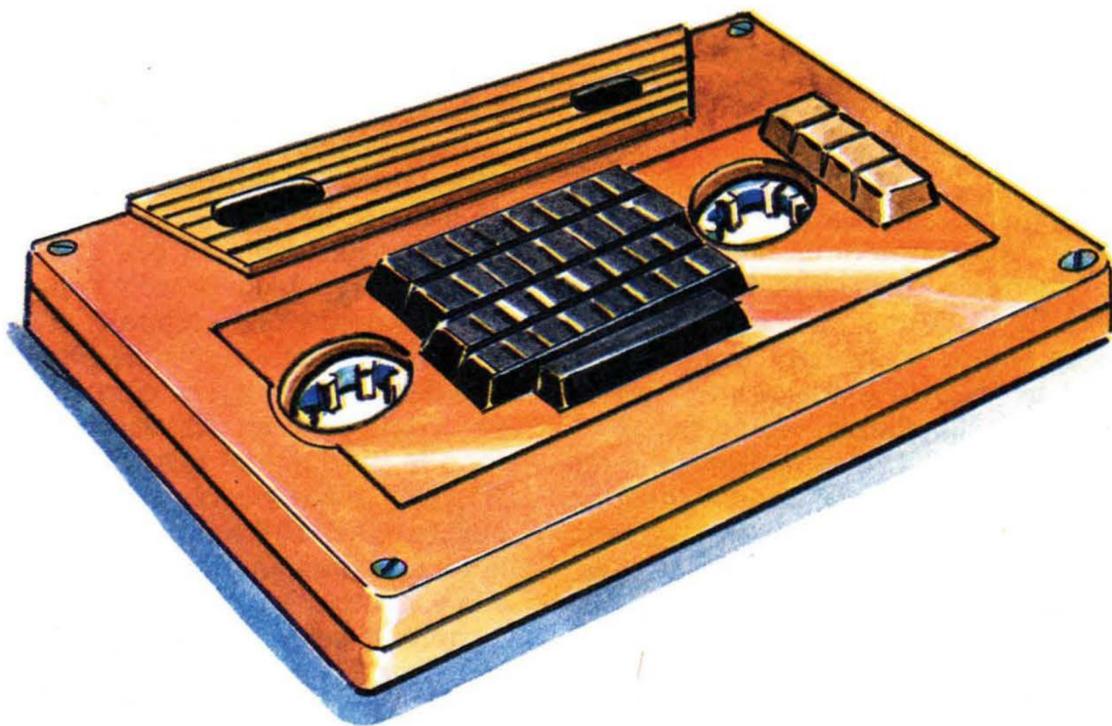
Risulta evidente che i file sequenziali sono molto più lenti dei relativi e se a questa lentezza sommiamo quella insita nelle operazioni con il registratore a cassette, il tempo necessario al trattamento di file mediamente lunghi sale notevolmente.

Fortunatamente esistono particolari tecniche, che permettono di ridurre notevolmente il tempo di accesso al singolo record fino a renderlo inferiore a quello dei file relativi, anche se il tempo

globale di accesso al file rimane inevitabilmente superiore.

Queste tecniche consistono nel caricare in RAM l'intero file all'inizio delle operazioni, quindi manipolarlo con gran velocità (tutte le operazioni che avvengono in RAM hanno una velocità enormemente superiore a qualsiasi operazione anche su disco), infine risalvare su nastro il file manipolato. I lunghi tempi necessari all'iniziale caricamento del file e al suo finale salvataggio, sono compensati dalla velocità delle operazioni sul file, siano esse l'aggiunta di un record, la sua ricerca o un ordinamento alfabetico. Il file esistente su nastro vie-

ne trasferito nella RAM del computer, all'interno di opportuni vettori dimensionati su un valore massimo, che rappresenta la capacità del file (massimo numero di record). Il contenuto di questi vettori può quindi essere alterato e manipolato in qualsiasi modo attraverso le opzioni del database, aggiungendo, eliminando o modificando qualsiasi record. Alla fine delle operazioni si potrà perciò risalvare il tutto su nastro. Poiché, come abbiamo detto, le operazioni più costose in termini di tempo sono quelle con l'unità a cassette, risulta assai conveniente caricare il file all'inizio dell'attività - generalmente al



mattino - e procedere al suo salvataggio solo alla sera, al termine dell'utilizzo del database. Per tutta la giornata lavorativa disporremo così di un archivio estremamente veloce e funzionale.

Utilizzo del programma

La struttura programma/utente è stata mantenuta simile a quella di "Refiling System 1.0", per agevolare chi desidera utilizzare entrambi i programmi. All'attivazione del programma viene richiesto di inserire la data corrente, nella forma giorno/mese/anno (esempio 011186), e l'ora corrente, nella forma ore/minuti/secondi (esempio 110800). Se non si desidera - per brevità - inserire alcun valore, si preme semplicemente RETURN al prompt del programma; in questo caso, tuttavia, non verranno registrate data ed ora dell'ultimo aggiornamento effettuato sul file dati.

Comparirà quindi un menu di selezione contenente le seguenti opzioni:

0. Crea nuovo file
1. Carica file
2. Salva file
3. Scrive record
4. Legge record
5. Ricerca record
6. Ordina record
7. Memoria libera
8. Stampa il file
9. Fine lavoro

Dal momento che alla prima utilizzazione non disporremo di un file dati su nastro, dovremo generarlo. Selezioniamo quindi l'opzione 0.

Il programma ci chiederà il numero di campi per record e quindi il loro nome e rispettiva lunghezza. Viene quindi richiesta la capacità del file (massimo numero di record): qui occorre prestare molta attenzione. Non esiste un tetto massimo a questo valore, in quanto esso dipende dal numero di campi che abbiamo stabilito precedentemente e dal tipo di espansione di memoria collegato al VIC 20: 100 record con 5 campi ciascuno occupano in genere più memoria di 80 record con 15 campi ciascuno... Il valore va quindi trovato sperimentalmente in rapporto alla memoria disponibile.

Ad ogni buon conto, dopo aver assegnato le dimensioni al file e quindi il suo nome, è conveniente controllare tramite l'opzione 7 del menu l'ammontare di memoria libera: quest'ultimo non dovrebbe mai essere inferiore a circa 1000 byte, al fine di lasciare il sufficiente spazio per le variabili utilizzate dal programma. Dopo un minimo di esperienza si sarà in grado di ottimizzare la capacità dei file sul massimo valore possibile.

Stabilite le massime dimensioni del file dovremo scegliere un nome da assegnargli: chiamiamolo "PROVA".

A questo punto potremo iniziare a inserire informazioni attraverso l'opzione 3: comparirà nella parte alta dello schermo una linea in campo inverso contenente

le opzioni usabili in modo inserimento dati. Esse sono:

- tasto "-" passa al record precedente
- tasto "+" passa al record successivo
- tasto "£" passa a un record particolare
- tasto "E" ritorna al menu principale
- tasto "I" permette l'inserimento dati

A questo punto saremo posizionati sul record 1. Premiamo il tasto I per l'inserimento dati. Comparirà il nome del primo campo e tra parentesi la sua lunghezza: e saremo pronti per scrivere i nostri dati. Se scrivendo raggiungiamo la fine del campo, il computer passerà automaticamente al campo successivo; se invece i dati che desideriamo inserire hanno una lunghezza inferiore alla massima lunghezza del campo, dovremo premere RETURN per passare al campo successivo.

Se non si desidera scrivere su un campo, è sufficiente premere RETURN. Dopo aver inserito i dati in tutti i campi il programma si posizionerà sul record successivo, in questo caso il numero 2.

Un record può essere riscritto in qualsiasi momento, posizionandosi su di esso con i tasti "+" e "-" oppure con il tasto "£" (lira), che permette di scrivere su qualunque record, comunicandone il numero al computer. Premendo quindi il tasto "I" si potrà riscrivere il record. Scriviamo anche i dati del record 2 e ritorniamo al menu principale, premendo il tasto "E".

Il nostro file conterrà ora due record, che potremo leggere con l'opzione 4. La schermata che compare è analoga a quella di scrittura (tranne per la parte alta dello schermo, che contiene, in rosso, i dati generali del file) e identiche sono le funzioni dei tasti tranne il tasto "I", che, ovviamente, non compare in questa opzione. Il programma si posizionerà automaticamente sul primo record e lo visualizzerà sullo schermo. Potremo leggere il secondo sia premendo "+" che premendo "£" e rispondendo con 2 alla domanda del computer.

Dal momento che abbiamo inserito soltanto due record, cercando di leggere un record non scritto, ad esempio il numero 3, il computer ci avviserà che tale record è vuoto.

Premendo "E" ritorniamo al menu principale.

Una delle caratteristiche più utili dei programmi di archivio è la possibilità di ricercare un certo record all'interno di un file, che magari ne contiene più di mille. L'opzione 5 esegue appunto questa funzione. Selezionando tale opzione, il computer visualizzerà l'elenco dei campi di ogni record e ci chiederà in base a quale campo desideriamo effettuare la ricerca. Ad esempio, volendo ricercare all'interno di un file di indirizzi composto da 5 campi (Cognome, Nome, Via, Città, Numero telefonico) il record contenente i dati del signor Rossi, sceglieremo di effettuare la ricerca sul primo campo, e digiteremo "Rossi" alla

richiesta circa il contenuto del campo. Il programma - dopo averci chiesto se desideriamo o meno una stampa dei dati - ricercherà quindi, all'interno del file, tutti i cognomi "Rossi" nel primo campo. Se questo cognome compare nell'archivio, verrà visualizzato il record contenente i suoi dati.

Il programma chiederà quindi se si desidera proseguire nella ricerca (ci possono essere più persone il cui cognome è Rossi). In caso affermativo la ricerca prosegue fino alla ricorrenza successiva o alla fine del file. Se invece si desiderasse controllare tutte le persone che abitano a Milano, si sceglierà per la ricerca il campo numero 4 e gli si assegnerà come contenuto la stringa "Milano".

Una particolarità interessante dell'opzione di ricerca, è che essa permette di ricercare un particolare record semplicemente digitando solo una parte del suo campo.

Per spiegarci meglio, se il file contiene un record per il signor Rossi, uno per il signor Rossetti ed uno per il signor Rosellini, ricercando "Rossi" verrà trovato solo il record del signor Rossi. Se alla domanda del computer sul contenuto del campo di ricerca digitiamo invece "Ross", il programma troverà sia "Rossi" che "Rossetti". Se digitiamo "Ros" verrà trovato anche "Rosellini". Se digitiamo "R" verranno trovati tutti i record che iniziano per R.

Questa particolarità della funzione di ricerca risulta estremamente utile, se non ricordiamo esattamente un cognome o la forma del dato che ci serve. Ritorniamo al menu principale. L'opzione 7 ci dice l'ammontare della memoria disponibile (dopo un tempo di attesa più o meno lungo, necessario al computer per effettuare i suoi controlli).

L'opzione 6 ordina invece alfabeticamente tutto il file, a partire dal primo campo e poi via via per tutti i campi successivi, fino all'ultimo carattere di ogni record. Il tempo di attesa per l'operazione di ordinamento dipende dal numero esistente di record, dalla loro lunghezza, e dal loro stato di "disordine" all'interno del file. Non è mai tuttavia troppo elevato, dal momento che tutta l'operazione avviene in RAM.

L'opzione 8 genera uno stampato di tutto il file, con l'ordine secondo il quale i record si trovano all'interno del file stesso. Ricordiamo che la stampa di record singoli è disponibile attraverso l'opzione di ricerca.

Infine le opzioni 1 e 2 servono rispettivamente per il caricamento o il salvataggio su nastro di un file dati. Viene in entrambi i casi richiesto il nome del file, il quale è tuttavia opzionale.

Si può uscire dal programma attraverso l'opzione 9, che dopo aver richiesto conferma provoca l'uscita dal programma.

Ecco ora un elenco delle principali variabili utilizzate dal programma:

RN massimo numero di record
 TY numero di record allocati
 F\$ nome del file dati
 RL lunghezza dei record
 C numero dei campi
 N\$() nome dei campi
 L() lunghezza dei campi
 RC\$() contenuto dei record
 R\$ ritorno carrello
 BY ammontare di byte liberi
 F variabile di flag
 A codice ASCII di un tasto
 D\$,U\$ data aggiornamento
 T1\$,T\$ ora aggiornamento
 R contatore dei record
 PS posizione di una sottostringa

per l'inserimento dei dati.
 370-480 definizione del file dati. Si definiscono il numero di campi per record, il loro nome e lunghezza, e la capacità e il nome del file. La massima lunghezza consentita di un record è di 255 caratteri (254 effettivi, in quanto il programma aggiunge automaticamente un CHR\$(13) di fine record).
 490-500 subroutine di attesa della pressione di un tasto. Viene restituito anche il codice ASCII del tasto premuto nella variabile A.
 520-700 subroutine di inserimento dati. Questa subroutine permette sia l'inserimento sequenziale dei record che il loro inserimento casuale all'interno del file, posto che non esistano altri record ancora vuoti, dotati di numero progressivo inferiore.
 720-840 subroutine di lettura dei record. Anch'essa permette sia la lettura sequenziale che quella di un record qualunque all'interno del file.
 940-970 subroutine di visualizzazione dei parametri principali del file durante la lettura dei record.
 990-1070 subroutine di salvataggio su nastro dell'intero file.

1090-1170 subroutine di caricamento di un file da nastro e di automatico dimensionamento dei vettori del programma.
 1180-1320 menu di selezione delle opzioni offerte dal programma.
 1340-1490 ricerca di un certo record all'interno del file, in base al contenuto di uno qualsiasi dei suoi campi. La ricerca avviene con il principio della "wild card".
 1510-1580 subroutine di ordinamento sequenziale e alfabetico del file. Vengono confrontati tra loro tutti i caratteri costituenti i vari record, e posti in ordine strettamente alfabetico.
 1600-1630 subroutine di stampa di tutti i record contenuti nel file.
 1670-1680 visualizzazione dell'ammontare della memoria libera. È opportuno che questo valore non scenda mai al di sotto di 1000, a meno che manchino solo pochi record al raggiungimento della massima capacità del file.
 1700-1710 fine delle operazioni. Il programma chiede conferma e, in caso affermativo, esce dal programma.

Commento al listato

100-110 inizializzazione colori schermo, bordo e testo e passaggio al modo maiuscolo/minuscolo.
 130-160 accettazione data ed ora corrente. A causa delle istruzioni CLR presenti nel programma, la data viene immagazzinata permanentemente nelle tre locazioni di memoria 667, 668 e 669 nel suo formato, rispettivamente, di giorno, mese e anno.
 180-350 subroutine di input controllato

Seqfiling System 1.0

```

100 REM **INIZIALIZZAZIONE GENERALE**
      :rem 51
110 PRINTCHR$(14) :rem 214
120 REM **LA DATA VIENE IMMAGAZZINATA NELLE
    LE LOCAZIONI 667, 668 E 669**
      :rem 127
130 INPUT"{CLR}DATA ({ 2 G}{ 2 M}{ 2 A}
    ){ 2 SPAZI}000000{ 8 SIN}";D$
      :rem 59
140 I=VAL(LEFT$(D$,2)):POKE667,I:I=VAL(MI
    D$(D$,3,2)):POKE668,I :rem 32
150 I=VAL(RIGHT$(D$,2)):POKE669,I
      :rem 140
160 INPUT"{GIU'}ORA{ 2 SPAZI}({ 2 H}
    { 2 M}{ 2 S}){ 2 SPAZI}000000
    { 8 SIN}";T1$:GOTO1180 :rem 64
170 REM **SUBROUTINE INPUT CONTROLLATO **
      :rem 21
180 PRINT"INSERISCI:" :F=1:B$="":LS=0:A$="
    ":A=0 :rem 42
190 GETA$:IFA$=""THEN190 :rem 87
200 A=ASC(A$+CHR$(0)) :rem 74
210 IFA$=CHR$(20) THENLS=LS-1:F=0:B$=LEFT$
    (B$,LEN(B$)-1):GOTO250 :rem 122
220 IFA$=CHR$(13) THENF=0:GOTO310:rem 100
230 IF(A>31ANDA<95)OR(A>159ANDA<233) THEN2
    50 :rem 212
240 PRINT:PRINT"CARA{ 2 T}ERE E{ 2 R}ATO
    !":PRINT:GOTO180 :rem 10
250 PRINTA$; :rem 196
260 IFF=1 THENB$=B$+A$:LS=LS+1:IFLEN(B$)=L
    (W) THENRETURN :rem 83
270 F=1:GOTO190 :rem 90
300 REM **AGGIUNGE SPAZI SHIFTATI FINO A
    LUNGHEZZA CAMPO** :rem 222
310 A=LEN(B$):D=L(W)-A:IFA=0 THENRETURN
      :rem 75
320 IFD>0 THEN340 :rem 156
330 IFD<0 THENB$=LEFT$(B$,L(W)):RETURN
      :rem 231
340 A$=CHR$(160):C$="":FORI=1 TOD:C$=C$+A$
    :NEXT:B$=B$+C$:RETURN :rem 64
350 FORI=1 TO3000:NEXT:RETURN :rem 47
360 REM **DEFINIZIONE DEL FILE** :rem 78
370 CLR:INPUT"{CLR}CAMPI PER RECORD";C:PR
    INT:DIMN$(C),L(C) :rem 23
380 RL=0:FORI=1 TOC :rem 100
390 PRINT"{RVS}CAMPO N.{OFF}"I:PRINT
      :rem 141
400 INPUT"NOME";N$(I) :rem 94
410 INPUT"LUNGHEZZA";L(I) :rem 194
420 RL=RL+L(I):PRINT:NEXT :rem 96
430 RL=RL+1:PRINT"LUNGHEZZA TOTALE DI UNR
    ECORD:";RL;"CARATT.":PRINT :rem 173
440 IFRL>255 THENPRINT"RECORD TRO{ 2 P}O
    LUNGI!":FORI=1 TO5000:NEXT:GOTO1180
      :rem 32
450 RN=0:INPUT"MAX. N.RECORD";RN:PRINT
      :rem 165
460 DIMRC$(RN) :rem 30
470 INPUT"NOME FILE";F$:IFLEN(F$)>0 ANDLEN
    (F$)<11 THENF$=CHR$(18)+F$:GOTO1180
      :rem 34
480 PRINT"DA 1 A 10 CARA{ 2 T}ERI!":GOTO4
    70 :rem 147
490 GETA$:IFA$=""THEN490 :rem 93
500 A=ASC(A$+CHR$(0)):RETURN :rem 103
510 REM **INSERIMENTO DI UN RECORD**
      :rem 94
520 R=TY+1 :rem 47
530 PRINT"{CLR}{RVS}+ SU{ 2 C}. - PREC. E
    SING{ 2 SPAZI}E ESCE{ 2 SPAZI}I INS
    ER.{ 4 SPAZI}" :rem 132
540 PRINT"RECORD #R :rem 158
550 PRINT"{GIU'}SCELTA?" :rem 247
560 GOSUB490:IFA<>43ANDA<>45ANDA<>92ANDA<
    >69ANDA<>73 THEN560 :rem 82
570 IFA=43 THENR=R+1 :rem 124

```

```

580 IFA=43ANDR>TY+1THENPRINT"ESISTONO REC
ORD VUOTI PIU'BASSI!":GOTO550
:rem 124
590 IFA=43THEN530 :rem 217
600 IFA=45THENR=R-1:IFR<1THENR=1:rem 241
610 IFA=45THEN530 :rem 212
620 IFA=92THENINPUT"NUMERO RECORD";R
:rem 181
630 IFA=92ANDR>TY+1THENPRINT"ESISTONO REC
ORD VUOTI PIU'BASSI!":GOTO620
:rem 122
640 IFA=92THEN530 :rem 217
650 IFA=69THEN1180 :rem 16
660 RC$(R)="":FORW=1TOC:PRINT:B$=""
:rem 45
670 PRINTN$(W)"("L(W)"CARATT.):":GOSUB180
:rem 139
680 RC$(R)=RC$(R)+B$ :rem 36
690 NEXT:IFR<=TYTHEN520 :rem 229
700 TY=TY+1:GOTO520 :rem 148
710 REM **LEGGE IL FILE** :rem 61
720 R=1 :rem 89
730 GOSUB940:PRINT"{RVS}+ SU{ 2 C}.
{ 3 SPAZI}- PREC.{ 5 SPAZI}E SING.
{ 3 SPAZI}E ESCE{ 6 SPAZI}" :rem 128
740 PRINT"RECORD #";R:PRINT :rem 162
750 IFRCS$(R)="":THENPRINT"RECORD VUOTO!":G
OTO780 :rem 23
760 PS=1:FORK=1TOC:PRINTN$(K)": "MID$(RC$(
R),PS,L(K)):PRINT:PS=PS+L(K):rem 230
770 NEXT:PRINT :rem 164
780 PRINT"{GIU'}_SCELTA?":GOSUB490
:rem 83
790 IFA<>43ANDA<>45ANDA<>92ANDA<>69THENGO
SUB490:GOTO790 :rem 157
800 IFA=43THENR=R+1:GOTO730 :rem 133
810 IFA=45THENR=R-1:IFR<1THENR=1:rem 244
820 IFA=45THEN730 :rem 217
830 IFA=92THENINPUT"N. RECORD";R:GOTO730
:rem 107
840 -A=69THEN1180 :rem 175
930 REM **DATI DEL FILE** :rem 63
940 PRINT"{CLR}{RED}FILE:"F$;TAB(22)"
{OFF}CAPACITA':"RN :rem 53
950 PRINT"CAMPI:"C;TAB(22)"RECORD:"TY
:rem 7
960 PRINT"AGGIORN.AL:"U$TAB(22)"ORE:"LEFT
$(T$,2)" "MID$(T$,3,2)" "RIGHT$(T$,2)
:rem 32
970 PRINT"{BLU}";:RETURN :rem 229
980 REM **SALVA IL FILE** :rem 89
990 GOSUB940 :rem 191
1000 PRINT"INSERISCI LA CASSETTA E PREMI
RETURN" :rem 66
1010 GOSUB490:IFA<>13THEN1010 :rem 184
1020 D$=STR$(PEEK(667))+STR$(PEEK(668))+S
TR$(PEEK(669)) :rem 86
1030 R$=CHR$(13):OPEN1,1,1,F$ :rem 26
1040 PRINT#1,RN;R$;TY;R$;C;R$;RL;R$;D$;R$
;TI$ :rem 197
1050 FORI=1TOC:PRINT#1,N$(I);R$;L(I):NEXT
:rem 232
1060 FORI=1TOTY:PRINT#1,RC$(I):NEXT
:rem 200
1070 CLOSE1:GOTO1180 :rem 172
1080 REM **CARICA IL FILE** :rem 173
1090 CLR:INPUT"{CLR} NOME DEL FILE";F$:F$
=CHR$(18)+F$ :rem 198
1100 OPEN1,1,0,F$ :rem 116

```

LOAD

SEQFILING
SYSTEM

```

1110 INPUT#1,RN,TY,C,RL,U$,T$:DIMN$(C),L(
C),RC$(RN) :rem 202
1120 FORI=1TOC:INPUT#1,N$(I),L(I):NEXT
:rem 41
1130 FORI=1TOTY:RC$(I)=" :rem 193
1140 GET#1,A$:RC$(I)=RC$(I)+A$:IFA$<>CHR$
(13)THEN1140 :rem 139
1150 NEXT :rem 6
1160 CLOSE1:GOTO1180 :rem 172
1170 REM **MENU DI SELEZIONE** :rem 197
1180 PRINT"{CLR}{RVS}SEQFILING 1.0
{ 3 SPAZI}MCG 85" :rem 181
1190 A$="{ 2 DES}" :rem 235
1200 PRINT"{GIU'}"A$"0. CREA NUOVO FILE"
:rem 254
1210 PRINTA$"1. CARICA FILE" :rem 156
1220 PRINTA$"2. SALVA FILE" :rem 114
1230 PRINTA$"3. SCRIVE RECORD" :rem 104
1240 PRINTA$"4. LEGGE RECORD" :rem 2
1250 PRINTA$"5. RICERCA RECORD" :rem 153
1260 PRINTA$"6. ORDINA RECORD" :rem 95
1270 PRINTA$"7. MEMORIA LIBERA" :rem 158
1280 PRINTA$"8. STAMPA IL FILE" :rem 98
1290 PRINTA$"9. FINE LAVORO" :rem 222
1300 PRINT"{ 2 GIU'}_SCELTA?" :rem 50
1310 GOSUB490:IFA<48ORA>57THEN1310
:rem 20
1320 ONA-47GOTO370,1090,990,520,720,1340,
1510,1670,1600,1700 :rem 35
1330 REM **RICERCA UN RECORD** :rem 174
1340 PRINT"{CLR}I CAMPI SONO:"FORI=1TOC:
PRINTI":";N$(I):NEXT :rem 228
1350 PRINT"{GIU'}NUMERO DEL CAMPO SU CUI
EFFETTUARE LA" :rem 71
1360 INPUT"RICERCA";A:IFA<1ORA>CTHEN1350
:rem 229
1370 PS=1:IFA>1THENFORI=1TOA-1:PS=PS+L(I)
:NEXT :rem 171
1380 INPUT"{GIU'}CONTENUTO";C$ :rem 146
1390 INPUT"VUOI UNO STAMPATO{ 2 SPAZI}N
{ 3 SIN}";S$:IFSS$="S"THENOPEN4,4,7:C
MD4 :rem 68
1400 FORI=1TOTY:IFC$<>MID$(RC$(I),PS,LEN(
C$))THEN1470 :rem 140
1410 PRINT"{CLR}RECORD N."I :rem 174
1420 PO=1:FORW=1TOC:IFPO=1THENPRINTN$(1)"
:"LEFT$(RC$(I),L(1)):GOTO1440
:rem 205
1430 PRINTN$(W)": "MID$(RC$(I),PO,L(W))
:rem 122
1440 PO=PO+L(W):NEXT :rem 220
1450 INPUT"{GIU'}PROSEGUO NELLA RICERCA (
S/N){ 2 SPAZI}S{ 3 SIN}";A$:rem 243

```

```

1460 IFA$<>"S"THEN1180 :rem 201
1470 NEXT :rem 11
1480 IFS$="S"THENPRINT#4:CLOSE4 :rem 158
1490 PRINT"{GIU'}ESITO NEGATIVO RICERCA.
PREMI UN TASTO":GOSUB490:GOTO1180
:rem 247
1500 REM **ORDINAMENTO RECORD** :rem 81
1510 PRINT"{CLR}ORDINAMENTO IN CORSO. ATT
ENDERE." :rem 128
1520 F=0 :rem 123
1530 FORI=1TOTY-1:IFRC$(I)<=RC$(I+1)THEN1
560 :rem 84
1540 A$=RC$(I):RC$(I)=RC$(I+1):RC$(I+1)=A
$ :rem 195
1550 F=1 :rem 127
1560 NEXT :rem 11
1570 IFF=1THEN1520 :rem 7
1580 GOTO1180 :rem 209
1590 REM **STAMPA IL FILE** :rem 214
1600 PRINT"{CLR}POSIZIONA LA CARTA SU STA
MPANTE E{DES}PREMI UN{ 2 SPAZI}TASTO
" :rem 193
1610 GOSUB490:OPEN4,4,7:CMD4:FORR=1TOTY
:rem 187
1620 PS=1:FORK=1TOC:PRINTN$(K)": "MID$(RC$(
R),PS,L(K)):PRINT:PS=PS+L(K)
:rem 18
1630 NEXT:PRINT:NEXT:PRINT#4:CLOSE4:GOTO1
180 :rem 136
1660 REM **MEMORIA LIBERA** :rem 18
1670 PRINT"{CLR}ATTENDERE...":BY=FRE(0)-(
FRE(0)<0)*65536 :rem 195
1680 PRINT"{CLR}"BY"BYTE LIBERI":GOSUB350
:GOTO1180 :rem 140
1690 REM **FINE OPERAZIONI** :rem 126
1700 INPUT"{CLR}CONFERMA{ 2 SPAZI}N
{ 3 SIN}";A$:IFA$<>"S"THEN1180
:rem 247
1710 END :rem 160

```



Novità firmate Jackson.

IL PERSONAL COMPUTER NELL'ORGANIZZAZIONE DELLE PICCOLE AZIENDE: APPLICAZIONI DEL MULTIPLAN
 COD. 578P L. 29.000

IL MANUALE DEL BASIC
 COD. 534A L. 45.000

AMBIENTE UNIX
 COD. 543P L. 19.000

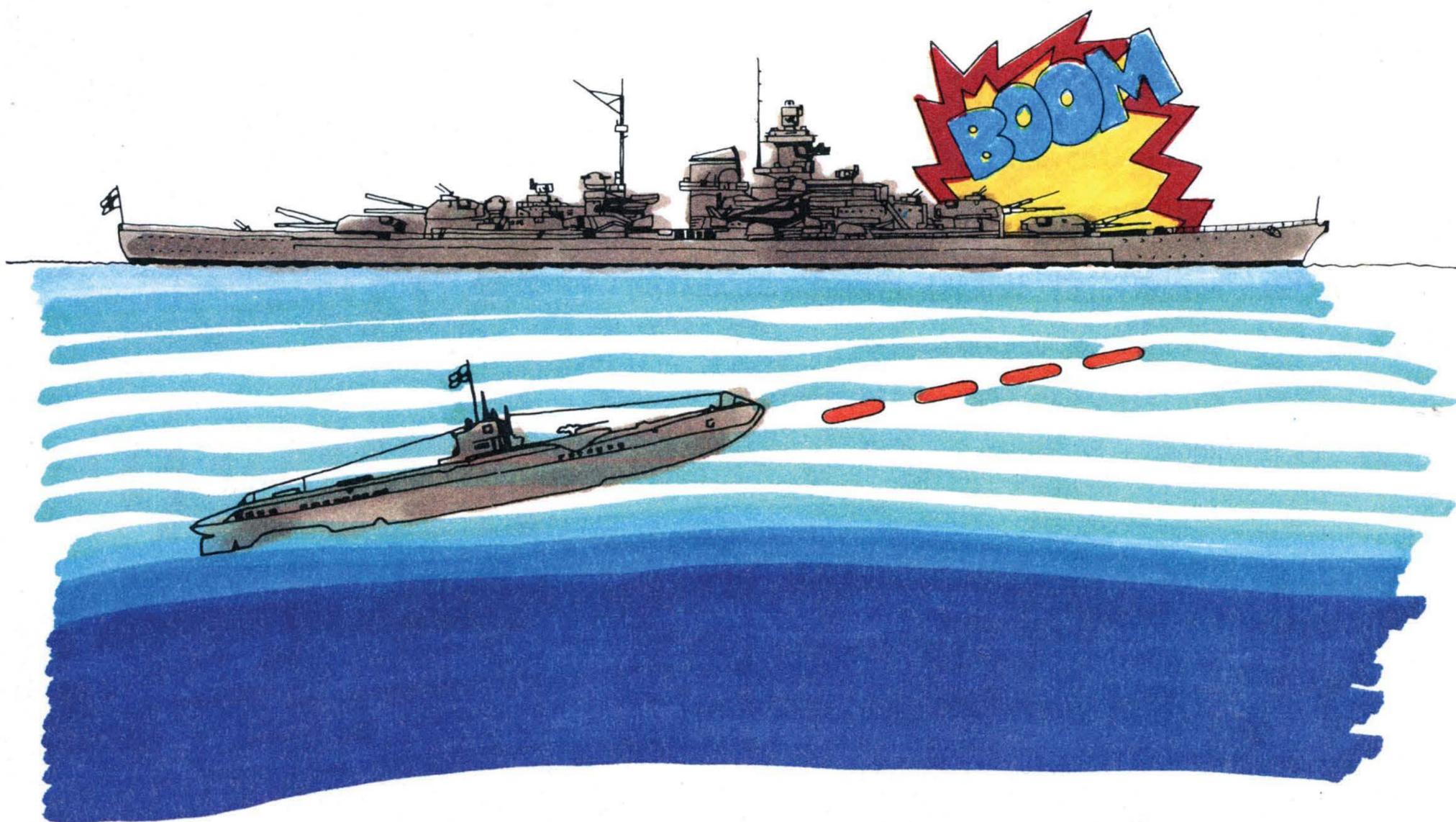
GRUPPO EDITORIALE JACKSON

La biblioteca che fa testo.



Battaglia navale

di **J. Butterfield**
trad. e adatt.
di **R. Luzzo**



Ecco un gioco in versioni per VIC 20 e C16-Plus/4 che farà rimpiangere a molti di non trovarsi più sui banchi di scuola. La versione per VIC 20 richiede un'espansione di memoria da almeno 3 Kbyte.

Il gioco della battaglia navale si può tranquillamente definire un classico: due scacchiere quadrettate vengono usate come campo di battaglia, sul quale i due giocatori dispongono le proprie

navi.

Poiché il vostro avversario sarà il computer, esso disporrà le proprie navi sulla scacchiera di sinistra - naturalmente senza mostrarvi la loro posizione -, mentre voi dovrete disporre le vostre sulla scacchiera di destra. Il computer è molto onesto e non sbircierà mai la posizione delle vostre navi (più che sbirciare, visto il soggetto, si dovrebbe parlare di PEEK...), che potranno essere disposte in qualunque posizione: attaccate ai bordi della scacchiera, adiacenti tra loro, o in diagonale; non è invece ovvia-

mente possibile sovrapporre tra loro due navi.

Sia voi che il computer disponete di una corazzata e di un incrociatore con lunghezza di 5 quadratini, un cacciatorpediniere di lunghezza 4, un sommergibile di lunghezza 3 e una corvetta (chiamata MAS nella versione per VIC 20) di lunghezza 2.

Il computer vi chiederà di inserire le vostre navi sulla scacchiera, digitando le loro coordinate di inizio e fine separate da un trattino (ad esempio: E4-E6). Qualsiasi valore non corretto verrà rifiu-

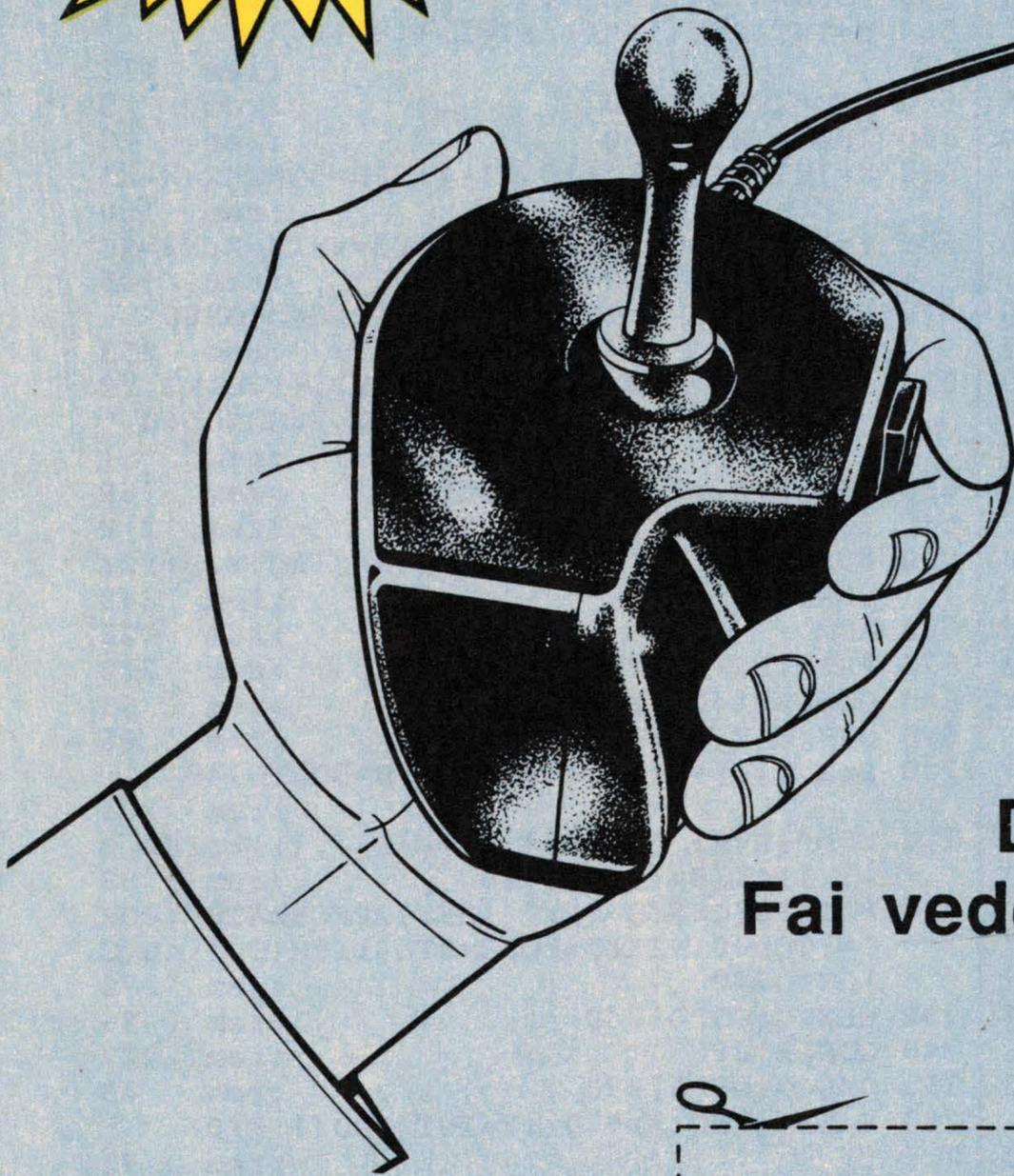
SPEEDKING

Joystick speciali per gente speciale
Dai campione!

46.000+
21.000=
55.000

Non è un errore, ma
la pacchia è solo fino al 30/5

Per il tuo C16 Joystick+
3 cassette giochi originali



Primo in

Tecnologia

Microswitch per la massima
precisione negli spostamenti
e la massima rapidità
di fuoco.

Design

Sta comodamente in mano.
Disponibile anche la
versione per mancini.

Qualità

Secondo il giudizio del
prestigioso mensile
"Commodore Horizons"
merita la media di 9/10:
precisione 9, robustezza 8,
facilità d'uso 10.
Il massimo!

Dai, campione!
Fai vedere chi sei con il tuo
Speedking.

GARANZIA MICROSTAR

Non perdere tempo.
Ritaglia e spedisce
oggi stesso a:

J.soft srl
Viale Restelli, 5
20124 Milano
Tel. 02/6880841-3

Tutti i prezzi comprendono
IVA e spese di spedizione

Pagherò al postino alla consegna

Nome Cognome

Indirizzo

Cod Fisc (solo se si desidera fattura)

Data

Inviatemi

- Speedking+3 cassette giochi originali
L.55.000 anziché L.67.000
(solo se spedisce entro il 30/5)

CAP/ Città

Firma

tato ed occorrerà ripetere l'inserimento.

La battaglia si svolge con un tiro ciascuno, semplicemente inserendo le coordinate del tiro stesso: sarà il computer che provvederà a segnalare sul video

l'effetto sortito, sia dai suoi che dai vostri colpi.

Con la massima onestà il computer vi avviserà se avete affondato una delle sue navi, in modo non dobbiate sprecare altri colpi attorno ad essa. La sua

strategia nel cercare di affondare la vostra flotta non è casuale, ma dotata di una certa "intelligenza artificiale", che gli permette di risultare un ottimo avversario.

Buon divertimento!

Battaglia navale versione per VIC 20

```
110 INPUT "{CLR} ISTRUZIONI"; Z$: IFLEFT$(Z$,
1) <> "S" GOTO 250 :rem 247
120 PRINT "{GIU'} ABBIAMO 5 NAVI A TESTA"
:rem 133
130 PRINT "NON SI POSSONO VEDERE LE NAVI D
EL NEMICO;" :rem 98
140 PRINT "SPARANDO SUL NEMICO," :rem 153
150 PRINT "UN COLPO A VUOTO VIENE"
:rem 220
160 PRINT "SEGNATO DA 'W', MENTRE" :rem 26
170 PRINT "UN CENTRO DA '{RVS} {OFF}'."
:rem 124
180 PRINT "INSERISCI LE NAVI CON" :rem 178
190 PRINT "LE COORD. DI INIZIO E" :rem 69
200 PRINT "FINE (ES. 'E4-E6')." :rem 57
220 PRINT "{GIU'} BUONA FORTUNA! PREMI
{ 2 SPAZI} UN TASTO PER" :rem 189
230 PRINT "INIZIARE..." :rem 75
240 GETZ$: IFZ$="" GOTO 240 :rem 139
250 DIM B(9,9,1), N$(4), L(4), H(4,1): R=TI
:rem 219
260 PRINT "{CLR} BATTAGLIA NAVALE" :rem 60
270 DATA CORAZZ., INCROC., CACCIA, SOMMERG.,
MAS :rem 19
280 DATA 5,5,4,3,2 :rem 103
290 FOR J=0 TO 4: READ N$(J): NEXT J: FORJ=0
TO 4: READ L(J): NEXT J :rem 229
300 R=R/1000: R1=RND(-R): IFR>1 GOTO 300
:rem 104
310 PRINT "LE MIE{ 7 SPAZI} LE TUE"
:rem 80
320 Z=0: Z$=" [<A>] [< 8 R>] [<S>]": GOSUB 1130
:rem 2
330 Z$=" [<Q>] { 8 +} [<W>]": FORZ=1 TO 8: GOSUB
1130: NEXT Z :rem 225
340 Z$=" [<Z>] [< 8 E>] [<X>]": GOSUB 1130
:rem 7
350 Z$="{HOME}{ 14 GIU'}" :rem 152
360 E$=Z$+" { 22 SPAZI}{SU}" :rem 189
370 PRINT "ABCDEFGH IJ" :rem 34
380 PRINT Z$;" POSIZIONA LE TUE NAVI"
:rem 142
390 FORJ=1 TO 3000: NEXTJ: PRINT "{SU} E DAMMI
LE COORDINATE" :rem 38
400 FORL=0 TO 1: FORJ=0 TO 9: FORK=0 TO 9: B(J,K,L
)=0: NEXTK,J,L :rem 150
410 FORL=0 TO 1: FORJ=0 TO 4: H(J,L)=0: NEXTJ,L
:rem 245
420 FORC=0 TO 4 :rem 4
430 REM INPUT COORDINATE :rem 243
438 PRINT E$; N$(C); " (L."; L(C); ")"
:rem 186
439 INPUT Y$: GOSUB 1140: IFE=1 GOTO 430
:rem 121
440 X1=X: Y1=Y: GOSUB 1150: IFE=1 GOTO 430
:rem 221
450 L=L(C): IFX<>X1 AND ABS(X-X1)<>L-1 GOTO 43
0 :rem 34
460 IFY<>Y1 AND ABS(Y-Y1)<>L-1 GOTO 430
:rem 132
470 IFY=Y1 AND X=X1 GOTO 430 :rem 11
480 Y2=SGN(Y1-Y): X2=SGN(X1-X): Y3=Y: X3=X
:rem 171
490 FORJ=1 TO L-1: IFB(X3,Y3,0)<>0 GOTO 430
:rem 254
500 IFX2*Y2<>0 THEN IFB(X3+X2,Y3,0)=B(X3,Y3
+Y2,0) THEN IFB(X3,Y3+Y2,0)<>0 GOTO 430
:rem 247
510 X3=X3+X2: Y3=Y3+Y2: NEXTJ :rem 166
520 IFB(X3,Y3,0)<>0 GOTO 430 :rem 210
530 PRINT Z$: FORI=1 TO 3: PRINT "{ 21 SPAZI}":
NEXT :rem 213
535 Y$="Q": FORJ=1 TO L: B(X,Y,0)=C+1: GOSUB 11
90: X=X+X2: Y=Y+Y2: NEXTJ,C :rem 136
540 PRINT E$;" SONO PRONTO!": S=0: H9=0
:rem 114
550 FORC=0 TO 4: L=L(C) :rem 171
560 D=INT(RND(1)*4): X1=L-1: Y1=L-1
:rem 113
570 IFD=2 THEN X1=0 :rem 3
580 IFD=0 THEN Y1=0 :rem 3
590 X=INT(RND(1)*(10-X1)): IFD=3 THEN X=X+X1
:rem 180
600 Y=INT(RND(1)*(10-Y1)): Y3=Y: X3=X: X2=SG
N(2-D): Y2=SGN(Y1) :rem 185
610 FORJ=1 TO L-1: IFB(X3,Y3,1)<>0 GOTO 560
:rem 253
620 IFX2*Y2<>0 THEN IFB(X3+X2,Y3,1)=B(X3,Y3
+Y2,1) THEN IFB(X3,Y3+Y2,1)<>0 GOTO 560
:rem 1
630 X3=X3+X2: Y3=Y3+Y2: NEXTJ :rem 169
640 IFB(X3,Y3,1)<>0 GOTO 560 :rem 218
650 FORJ=1 TO L: B(X,Y,1)=C+1: X=X+X2: Y=Y+Y2:
NEXTJ,C :rem 249
660 REM :rem 128
670 PRINT E$;: INPUT "TIRO"; Y$ :rem 210
680 GOSUB 1140: IFE=1 GOTO 660 :rem 53
690 B=B(X,Y,1): IFB>9 GOTO 660 :rem 66
700 B(X,Y,1)=B+10: Y$="W": IFE=0 GOTO 740
:rem 202
710 Y$="{RVS} ": H(B-1,1)=H(B-1,1)+1: IFH(B
-1,1)<L(B-1) GOTO 740 :rem 83
720 PRINT E$; N$(B-1); " {RVS} AFFONDATO": FOR
C=1 TO 500: NEXTC: FORC=0 TO 4: IFH(C,1)<L(C
) GOTO 740 :rem 176
730 NEXTC: GOTO 1240 :rem 86
740 GOSUB 1200 :rem 222
750 ONSGOTO 810,890 :rem 43
760 X=INT(RND(1)*10): Y=INT(RND(1)*10)
:rem 92
770 B=B(X,Y,0): IFB<9 GOTO 950 :rem 64
780 X=X+3: IFX<10 GOTO 770 :rem 127
790 X=X-10: Y=Y+1: IFY<10 GOTO 770 :rem 54
800 Y=Y-10: GOTO 770 :rem 38
810 X1=X9-D: X2=X9+D: Y1=Y9-D: Y2=Y9+D
:rem 105
820 IFX1<0 THEN X1=0 :rem 67
830 IFY1<0 THEN Y1=0 :rem 70
840 IFX2>9 THEN X2=9 :rem 91
850 IFY2>9 THEN Y2=9 :rem 94
860 FORX=X1 TO X2: FORY=Y1 TO Y2 :rem 63
870 B=B(X,Y,0): IFB<9 GOTO 950 :rem 65
```

```

880 NEXTY,X:D=D+1:GOTO810 :rem 35
890 X=X9:Y=Y9 :rem 41
900 X=X+X8:Y=Y+Y8:IFX<0ORX>9ORY<0ORY>9GOT
O930 :rem 207
910 B=B(X,Y,0):IFB<9GOTO950 :rem 60
920 B=B-10:IFB>0GOTO900 :rem 53
930 IFS1=1THENS=1:GOTO810 :rem 25
940 S1=1:X8=-X8:Y8=-Y8:GOTO890 :rem 45
950 B(X,Y,0)=B+10:Y$="W":H7=0:H8=SGN(B):H
9=H9+H8:X$="ACQUA":IFB=0GOTO980 :rem 52
960 B=B-1:Y$="{RVS} ":X$="CENTRO":H(B,0)=
H(B,0)+1:IFH(B,0)<L(B)GOTO980 :rem 64
970 H7=1:H9=H9-L(B):X$="AFF. "+N$(B) :rem 116
980 GOSUB1190:PRINTE$;"IO ";X$;" IN ";CHR
$(X+65);CHR$(Y+48):FORJ=1TO1000:NEXTJ :rem 248
990 FORJ=0TO4:IFH(J,0)<L(J)GOTO1010 :rem 61
1000 NEXTJ:GOTO1250 :rem 133
1010 ONSGOTO1040,1090 :rem 166
1020 IFH8=1THENS=1:D=1:X9=X:Y9=Y:rem 222
1030 GOTO670 :rem 154
1040 IFH9=0THENS=0:GOTO670 :rem 65
1050 IFH8=0GOTO670 :rem 24
1060 X8=X-X9:Y8=Y-Y9:S1=0 :rem 245
1070 IFX8=SGN(X8)ANDY8=SGN(Y8)THENS=2 :rem 73
1080 GOTO670 :rem 159
1090 IFH7=0GOTO670 :rem 27
1100 S=1 :rem 131
1110 IFH9=0THENS=0 :rem 47
1120 GOTO670 :rem 154
1130 PRINTZ$;CHR$(Z+48);Z$:RETURN :rem 33
1140 P=1:E=0 :rem 112
1150 FOR Z=PTOLEN(Y$)-1:X=ASC(MID$(Y$,Z)
-65:IFX>=0ANDX<=9GOTO1170 :rem 207
1160 NEXTZ:E=1:RETURN :rem 104
1170 FOR P=ZTOLEN(Y$):Y=ASC(MID$(Y$,P))-4
8:IFY>=0ANDY<=9THENRETURN :rem 122
1180 NEXTP:GOTO1160 :rem 148
1190 PRINT"{HOME}";TAB(11);:GOTO1210 :rem 230
1200 PRINT"{HOME}"; :rem 226
1210 FORQ=1TOY+2:PRINT"{GIU'}";:NEXT Q :rem 228
1220 FORQ=1TOX+1:PRINT"{DES}";:NEXT Q :rem 239
1230 PRINT"{SIN}";Y$:RETURN :rem 6
1240 GOSUB1200:PRINTE$;"HAI VINTO":GOTO13
00 :rem 82
1250 PRINT:PRINT"HO VINTO! ECCO LE MIE NA
VI..." :rem 230
1260 FORX=0TO9:FORY=0TO9 :rem 20
1270 B=B(X,Y,1):IFB=0ORB>9GOTO1290 :rem 237
1280 Y$="Q":GOSUB1200 :rem 23
1290 NEXTY:NEXTX :rem 53
1300 PRINTZ$;"{ 5 GIU'}ANCORA"; :rem 146
1310 INPUTY$:IFLEFT$(Y$,1)<>"N"THENRUN
:rem 69
1) <>"S"GOTO250 :rem 247
120 PRINT"{GIU'}OGNUNO DI NOI POSSIEDE 5
NAVI SULLA" :rem 254
130 PRINT"GRIGLIA. NON SI POSSONO VEDERE
LE NAVI" :rem 196
140 PRINT"DELL'AVVERSARIO; SPARANDO SULLA
GRIGLIA" :rem 195
150 PRINT"DEL NEMICO, UN COLPO A VUOTO VI
ENE MO-" :rem 97
160 PRINT"STRATO COME W, MENTRE UN COLPO
A SEGNO" :rem 20
170 PRINT"VIENE MOSTRATO COME {RVS} {OFF}
." :rem 79
180 PRINT"INSERISCI LE COORDINATE DI INIZ
IO E DI" :rem 189
190 PRINT"FINE NAVE, AD ESEMPIO UN SOMMER
GIBILE" :rem 156
200 PRINT"(LUNGHEZZA 3) PUO' ESSERE POSIZ
IONATO" :rem 224
210 PRINT"CON 'E4-E6'." :rem 225
220 PRINT"{GIU'}BUONA FORTUNA! PREMI UN T
ASTO PER" :rem 189
230 PRINT"INIZIARE..." :rem 75
240 GETZ$:IFZ$=""GOTO240 :rem 139
250 DIM B(9,9,1),N$(4),L(4),H(4,1):R=TI
:rem 219
260 PRINT"{CLR}BATTAGLIA NAVALE":rem 60
270 DATA CORAZZATA,INCROCIATORE,CACCIA,SO
MMERGIBILE,CORVETTA :rem 15
280 DATA 5,5,4,3,2 :rem 103
290 FOR J=0 TO 4:READ N$(J):NEXT J:FORJ=0
TO4:READL(J):NEXTJ :rem 229
300 R=R/1000:R1=RND(-R):IFR>1GOTO300 :rem 104
310 PRINT "LE MIE{ 7 SPAZI}LE TUE" :rem 80
320 Z=0:Z$=" [<A>] [< 8 R>] [<S>]":GOSUB1130 :rem 2
330 Z$=" [<Q>] { 8 +} [<W>]":FORZ=1TO8:GOSUB
1130:NEXT Z :rem 225
340 Z$=" [<Z>] [< 8 E>] [<X>]":GOSUB1130 :rem 7
350 Z$="{HOME}{ 15 GIU'}" :rem 169
360 E$=Z$+"{ 40 SPAZI}{SU}" :rem 189
370 PRINT"↑↑↑↑↑↑↑↑↑↑":PRINT"ABCDEFGH IJ" :rem 217
380 PRINTZ$;"POSIZIONA LE TUE NAVI..." :rem 24
390 FORJ=1TO3000:NEXTJ:PRINTE$;"DAMMI LE
COORDINATE DI OGNI NAVE..." :rem 98
400 FORL=0TO1:FORJ=0TO9:FORK=0TO9:B(J,K,L)
=0:NEXTK,J,L :rem 150
410 FORL=0TO1:FORJ=0TO4:H(J,L)=0:NEXTJ,L :rem 245
420 FORC=0TO4 :rem 4
430 PRINTE$;N$(C);" (LUNGH.";L(C);")";:IN
PUTY$:GOSUB1140:IFE=1GOTO430:rem 50
440 X1=X:Y1=Y:GOSUB1150:IFE=1GOTO430 :rem 221
450 L=L(C):IFX<>X1ANDABS(X-X1)<>L-1GOTO43
0 :rem 34
460 IFY<>Y1ANDABS(Y-Y1)<>L-1GOTO430 :rem 132
470 IFY=Y1ANDX=X1GOTO430 :rem 11
480 Y2=SGN(Y1-Y):X2=SGN(X1-X):Y3=Y:X3=X :rem 171
490 FORJ=1TOL-1:IFB(X3,Y3,0)<>0GOTO430 :rem 254
500 IFX2*Y2<>0THENIFB(X3+X2,Y3,0)=B(X3,Y3

```

Battaglia navale versione per C16-Plus/4

```

100 COLOR0,1:COLOR4,1:COLOR1,2 :rem 47
110 INPUT"{CLR}ISTRUZIONI";Z$:IFLEFT$(Z$,

```

LOAD



BATTAGLIA
NAVALE

```

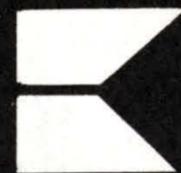
+Y2,0) THENIFB(X3,Y3+Y2,0)<>0GOTO430
:rem 247
510 X3=X3+X2:Y3=Y3+Y2:NEXTJ :rem 166
520 IFB(X3,Y3,0)<>0GOTO430 :rem 210
530 Y$="Q":FORJ=1TOL:B(X,Y,0)=C+1:GOSUB11
90:X=X+X2:Y=Y+Y2:NEXTJ,C :rem 131
540 PRINTE$;"SONO PRONTO!":S=0:H9=0
:rem 114
550 FORC=0TO4:L=L(C) :rem 171
560 D=INT(RND(1)*4):X1=L-1:Y1=L-1
:rem 113
570 IFD=2THENX1=0 :rem 3
580 IFD=0THENY1=0 :rem 3
590 X=INT(RND(1)*(10-X1)):IFD=3THENX=X+X1
:rem 180
600 Y=INT(RND(1)*(10-Y1)):Y3=Y:X3=X:X2=SG
N(2-D):Y2=SGN(Y1) :rem 185
610 FORJ=1TOL-1:IFB(X3,Y3,1)<>0GOTO560
:rem 253
620 IFX2*Y2<>0THENIFB(X3+X2,Y3,1)=B(X3,Y3
+Y2,1) THENIFB(X3,Y3+Y2,1)<>0GOTO560
:rem 1
630 X3=X3+X2:Y3=Y3+Y2:NEXTJ :rem 169
640 IFB(X3,Y3,1)<>0GOTO560 :rem 218
650 FORJ=1TOL:B(X,Y,1)=C+1:X=X+X2:Y=Y+Y2:
NEXTJ,C :rem 249
660 PRINTE$; :rem 205
670 INPUT".TIRO";Y$ :rem 149
680 GOSUB1140:IFE=1GOTO660 :rem 53
690 B=B(X,Y,1):IFB>9GOTO660 :rem 66
700 B(X,Y,1)=B+10:Y$="W":IFB=0GOTO740
:rem 202
710 Y$="{RVS} ":H(B-1,1)=H(B-1,1)+1:IFH(B
-1,1)<L(B-1)GOTO740 :rem 83
720 PRINTE$;N$(B-1);" {RVS}AFFONDATO":FOR
C=1TO500:NEXTC:FORC=0TO4:IFH(C,1)<L(C
)GOTO740 :rem 176
730 NEXTC:GOTO1240 :rem 86
740 GOSUB1200 :rem 222
750 ONSGOTO810,890 :rem 43
760 X=INT(RND(1)*10):Y=INT(RND(1)*10)
:rem 92
770 B=B(X,Y,0):IFB<9GOTO950 :rem 64
780 X=X+3:IFX<10GOTO770 :rem 127
790 X=X-10:Y=Y+1:IFY<10GOTO770 :rem 54
800 Y=Y-10:GOTO770 :rem 38
810 X1=X9-D:X2=X9+D:Y1=Y9-D:Y2=Y9+D
:rem 105
820 IFX1<0THENX1=0 :rem 67
830 IFY1<0THENY1=0 :rem 70
840 IFX2>9THENX2=9 :rem 91
850 IFY2>9THENY2=9 :rem 94
860 FORX=X1TOX2:FORY=Y1TOY2 :rem 63
870 B=B(X,Y,0):IFB<9GOTO950 :rem 65
880 NEXTY,X:D=D+1:GOTO810 :rem 35
890 X=X9:Y=Y9 :rem 41
900 X=X+X8:Y=Y+Y8:IFX<0ORX>9ORY<0ORY>9GOT
O930 :rem 207

```

```

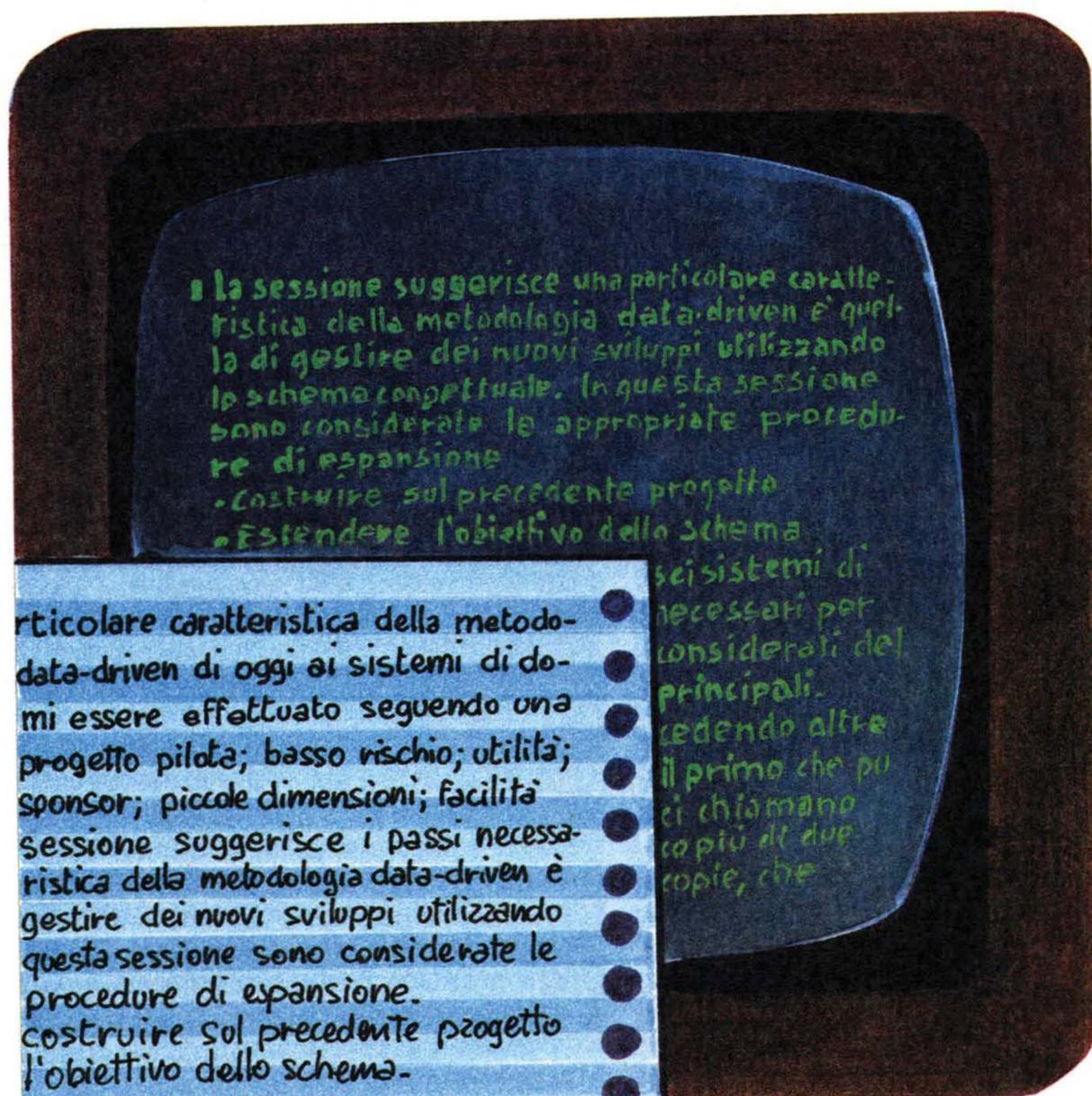
910 B=B(X,Y,0):IFB<9GOTO950 :rem 60
920 B=B-10:IFB>0GOTO900 :rem 53
930 IFS1=1THENS=1:GOTO810 :rem 25
940 S1=1:X8=-X8:Y8=-Y8:GOTO890 :rem 45
950 B(X,Y,0)=B+10:Y$="W":H7=0:H8=SGN(B):H
9=H9+H8:X$="ACQUA":IFB=0GOTO980
:rem 52
960 B=B-1:Y$="{RVS} ":X$="CENTRO":H(B,0)=
H(B,0)+1:IFH(B,0)<L(B)GOTO980
:rem 64
970 H7=1:H9=H9-L(B):X$="AFFONDO "+N$(B)
:rem 118
980 GOSUB1190:PRINTE$;"IO ";X$;" IN ";CHR
$(X+65);CHR$(Y+48);:FORJ=1TO1000:NEXT
J :rem 51
990 FORJ=0TO4:IFH(J,0)<L(J)GOTO1010
:rem 61
1000 NEXTJ:GOTO1250 :rem 133
1010 ONSGOTO1040,1090 :rem 166
1020 IFH8=1THENS=1:D=1:X9=X:Y9=Y:rem 222
1030 GOTO670 :rem 154
1040 IFH9=0THENS=0:GOTO670 :rem 65
1050 IFH8=0GOTO670 :rem 24
1060 X8=X-X9:Y8=Y-Y9:S1=0 :rem 245
1070 IFX8=SGN(X8)ANDY8=SGN(Y8)THENS=2
:rem 73
1080 GOTO670 :rem 159
1090 IFH7=0GOTO670 :rem 27
1100 S=1 :rem 131
1110 IFH9=0THENS=0 :rem 47
1120 GOTO670 :rem 154
1130 PRINTZ$;"-";CHR$(Z+48);" ";Z$:RETURN
:rem 126
1140 P=1:E=0 :rem 112
1150 FOR Z=PTOLEN(Y$)-1:X=ASC(MID$(Y$,Z))
-65:IFX>=0ANDX<=9GOTO1170 :rem 207
1160 NEXTZ:E=1:RETURN :rem 104
1170 FOR P=ZTOLEN(Y$):Y=ASC(MID$(Y$,P))-4
8:IFY>=0ANDY<=9THENRETURN :rem 122
1180 NEXTP:GOTO1160 :rem 148
1190 PRINT"{HOME}";TAB(13);:GOTO1210
:rem 232
1200 PRINT"{HOME}"; :rem 226
1210 FORQ=1TOY+2:PRINT"{GIU'}";:NEXT Q
:rem 228
1220 FORQ=1TOX+1:PRINT"{DES}";:NEXT Q
:rem 239
1230 PRINT"{SIN}";Y$:RETURN :rem 6
1240 GOSUB1200:PRINTE$;"HAI VINTO":GOTO13
00 :rem 82
1250 PRINT:PRINT"HO VINTO! ECCO LE MIE NA
VI..." :rem 230
1260 FORX=0TO9:FORY=0TO9 :rem 20
1270 B=B(X,Y,1):IFB=0ORB>9GOTO1290
:rem 237
1280 Y$="Q":GOSUB1200 :rem 23
1290 NEXTY:NEXTX :rem 53
1300 PRINTZ$;"{ 4 GIU'} ANCORA";:rem 129
1310 INPUTY$:IFLEFT$(Y$,1)<>"N"THENRUN
:rem 69

```



SuperCalc: un foglio elettronico per C16

di **M. Tinglof**
trad. e adatt.
di **M. Cristuib Grizzi**



La sessione suggerisce una particolare caratteristica della metodologia data-driven e quella di gestire dei nuovi sviluppi utilizzando lo schema concettuale. In questa sessione sono considerate le appropriate procedure di espansione.

- Costruire sul precedente progetto
- Estendere l'obiettivo dello schema

Particolare caratteristica della metodologia data-driven di oggi ai sistemi di domini essere effettuato seguendo una progetto pilota; basso rischio; utilità; sponsor; piccole dimensioni; facilità. La sessione suggerisce i passi necessari. La caratteristica della metodologia data-driven è gestire dei nuovi sviluppi utilizzando questa sessione sono considerate le procedure di espansione. Costruire sul precedente progetto l'obiettivo dello schema.

Sei sistemi di necessari per considerati dei principali. Vedendo altre il primo che può ci chiamano copio di due copie, che

Ecco un programma che rende uno scherzo l'effettuazione di complicati calcoli previsionali, o più semplicemente la piccola e comune contabilità domestica.

Quasi tutti sanno esattamente che cosa sia un word processor o un database, mentre il concetto di "foglio elettronico" rimane per lo più vago o del tutto sconosciuto.

Lo spreadsheet - nome inglese del foglio elettronico - altro non è che la versione informatica di un normale foglio di carta, usato in contabilità per incolonnare cifre e numeri.

Uno spreadsheet è composto da una serie di *celle*, ognuna delle quali contiene un dato. Le *celle* sono individuate, con il principio delle coordinate cartesiane, dal loro numero di riga e di colonna. Il numero contenuto nella cella

(3,10) sarà quindi il decimo valore della riga 3 o, in modo equivalente, il terzo valore della colonna 10. Ciò equivale a dire che il numero contenuto nella cella (3,10) si troverà in corrispondenza dell'intersezione tra la riga 3 e la colonna 10.

Per spiegarci meglio, ricorriamo a titolo di esempio a un "micro-spreadsheet", composto da 4 righe per 6 colonne. Esso potrà avere il seguente aspetto:

	Col.1	Col.2	Col.3	Col.4	Col.5	Col.6
Riga 1:	100	55	0	0	0	0
Riga 2:	250	125	0	0	0	0
Riga 3:	50	350	0	0	0	0
Riga 4:	500	200	0	0	0	0
Riga 5:	0	0	0	0	0	0

Nell'esempio i dati sono stati inseriti soltanto nelle colonne 1 e 2; la cella (1,1) contiene il valore 100, la (1,2) il valore 55, la (2,1) il valore 250 e così via... Rispetto al tradizionale foglio di carta per contabilità, uno spreadsheet offre molteplici vantaggi: tra questi possiamo elencare:

- totale precisione nei calcoli: il rischio di commettere errori in calcoli lunghi e ripetitivi viene completamente eliminato dall'infallibilità del computer in queste operazioni;
- estrema facilità di modifiche all'interno del foglio: per modificare un numero in una cella basta ridigitarlo secondo le proprie esigenze;
- riutilizzabilità: una volta definite le formule di calcolo che ci occorrono, è semplicissimo compiere delle analisi cosiddette *what if?* (cosa succede se?...), modificando i parametri e ricalcolando i totali.

	Capitale	Tasso/100	Ricavo	Guadagno
Dep.bancario	10000000	0.105	11050000	1050000
B.O.T.	30000000	0.128	33840000	3840000
Azioni S.p.A.	25000000	0.012	25300000	300000
C.C.T.	1000000	0.100	1100000	100000
Totale	66000000	0.345	71290000	5290000

Si noti che i tassi di interesse per le rispettive opzioni sono casuali, e che il totale della loro colonna è privo di significato.

Per applicare questo modello a SuperCalc occorrerà definire quattro colonne (Capitale, Tasso, Ricavo, Guadagno) e cinque righe (Dep.bancario, B.O.T., Azioni S.p.A., C.C.T. e Totale). Inoltre occorrerà definire le relazioni matematiche che intercorrono tra i valori: la colonna Ricavo è definita come il prodotto del Capitale per il Tasso, sommato al Capitale. La colonna Guadagno è definita come il prodotto tra il Capitale e il Tasso, mentre la riga Totale è definita come somma di tutti i numeri di ciascuna colonna.

Una volta che siano state definite queste correlazioni tra i dati sarà facile effettuare analisi del tipo "Cosa se?", semplicemente modificando a piacere i valori nello spreadsheet, in quanto il programma ricalcolerà automaticamente i valori interdipendenti. Ad esempio, se si decidesse di investire in B.O.T. venticinque milioni invece di trenta, basterà cambiare questo valore per avere istantaneamente ricalcolati guadagni e totali.

Dando il RUN a SuperCalc, apparirà

Uno spreadsheet può essere utilizzato per molteplici scopi, come il calcolo di budget, previsioni di investimento, calcolo di profitti e perdite, resoconti di spese sostenute e comunque per tutti quei compiti che implicano l'uso intensivo di dati numerici.

Utilizzare SuperCalc

L'uso del programma è semplice: nella maggior parte dei casi sarà esso stesso a chiedervi i dati di cui abbisogna. L'unica parte che richiede attenzione è la definizione iniziale del modello di calcolo dello spreadsheet. Dal momento che il sistema migliore per imparare è - come sempre - la sperimentazione, vedremo di passare in rassegna le possibilità di SuperCalc attraverso un esempio pratico.

Consideriamo il seguente modello per quattro diversi tipi di investimento:

una lista di comandi nella parte bassa dello schermo: è questo il *modo di operazione normale* e se fosse stato precedentemente definito un modello, questo verrebbe visualizzato al di sopra della lista di comandi. Accingiamoci quindi a definire il nostro semplicissimo modello.

Definire un modello

Entriamo in *Modo Programma* premendo il tasto P. Lo schermo si pulisce, una lista di numeri appare nella sua parte sinistra e una nuova lista di comandi viene visualizzata in basso. La lista di numeri a sinistra rappresenta un insieme di numeri di linea: sarà in queste linee che immagazzineremo la definizione del nostro modello.

A questo scopo usiamo il comando *Inserimento* premendo il tasto I. Il computer ci chiederà prima di quale linea desideriamo inserire il testo: rispondiamo con 1. Questa linea ora contiene un END, che rappresenta l'ultima istruzione nella definizione del modello.

Il computer chiederà quindi quante linee desideriamo inserire: rispondiamo con 12.

Righe e colonne

A partire dalla linea 1 fino alla 12 il

computer attenderà l'inserimento di dati per ciascuna linea. Per la linea 1 digitate:

1 NAME RA=DEPBNC

Il numero di linea (1) non deve - per precisione - essere digitato, in quanto è generato automaticamente dal programma. Questa linea assegna alla prima riga del nostro esempio il nome DEPBNC, contrazione di "Depositi Bancari", e sarà in seguito visualizzata nella parte sinistra dello schermo quando ritorneremo nel modo di operazione normale.

Nel termine RA, R indica che stiamo assegnando il nome a una riga, mentre A indica che si tratta della riga A, cioè la prima.

Sono ammessi al massimo sei caratteri per il nome di una riga.

Esistono in totale 20 righe, ognuna delle quali è contraddistinta da una lettera dell'alfabeto. Ora inserite:

2 NAME RB=BOT
3 NAME RC=AZIONI
4 NAME RD=CCT
5 NAME RE=TOTALE

Abbiamo così assegnato un nome a tutte e cinque le righe del nostro modello. Possiamo ora iniziare ad assegnare i nomi alle colonne, digitando:

6 NAME C1=CAPITALE,8

Questa istruzione assegna alla prima colonna, la numero 1, il nome CAPITALE. Anche qui C sta per colonna, mentre il ",8" ne definisce la larghezza di otto caratteri (sufficienti quindi a contenere un numero di otto cifre). Se non viene assegnata alcuna larghezza, il programma assume per default il valore 6 (un consiglio per risparmiare memoria e... nascondere l'inflazione: inserite nei vostri modelli i dati in unità di 1000 lire!).

Ogni elemento nella colonna, compreso il suo stesso nome, potrà dunque avere nel nostro esempio una lunghezza massima di otto caratteri.

Continuate gli inserimenti con:

7 NAME C2=TASSO,5
8 NAME C3=R1CAV0,8
9 NAME C4=GUADAGNO,8

Il prossimo passo sarà la definizione delle *formule* che permettono il calcolo dei valori richiesti.

La definizione delle formule

La prossima linea che dovremo definire potrà sembrare un po' insolita: è la prima espressione matematica che usiamo, e il suo scopo è definire la colonna RICAVO uguale al prodotto delle colonne CAPITALE e TASSO, sommato alla colonna CAPITALE.

Inserite in questo modo la linea 10:

10 DEFINE@3=@ 1*@ 2 +@ 1

Innanzitutto, ogni elemento (cella) nella matrice di righe e colonne possiede una sua univocità: ad esempio la cella C2

sarà data dall'incrocio della riga C con la colonna 2.

Il comando DEFINE pone il contenuto di una cella uguale a un'espressione contenente altri elementi ed eventuali costanti (come ad esempio 2 o 0.56). Può essere usata una qualsiasi delle quattro operazioni fondamentali (addizione, sottrazione, moltiplicazione e divisione), anche se non viene seguita alcuna priorità nell'esecuzione delle operazioni.

In un comando di definizione può risultare utile considerare più di una singola cella per volta, e ciò si ottiene usando il simbolo@.

@ può essere usato per sostituire la designazione sia di una riga che di una colonna. Per esempio, nel comando DEFINE di cui sopra @ 3 indica tutte le righe della colonna 3: esso potrebbe essere tradotto a parole con "Ogni riga della colonna 3 equivale a ogni riga della colonna 1 per ogni riga della colonna 2, sommata a ogni riga della colonna 1".

Quando inserite questo genere di comandi, assicuratevi che il carattere @ preceda il numero di colonna e segua la lettera corrispondente alla riga. Quando viene eseguita un'istruzione DEFINE, viene per prima cosa considerata la prima cella presente nell'istruzione: se è presente il simbolo@, viene eseguito un ciclo, in modo che ogni riga (o colonna, in dipendenza al formato usato nell'istruzione) venga calcolata secondo la formula contenuta nella DEFINE stessa.

Per esempio, incontrando @ 3, il programma considera ogni colonna nel ciclo come colonna corrente, calcola l'espressione e assegna il risultato alla colonna corrente. Al contrario, se il carattere @ compare all'interno della cella, viene sostituito dal valore corrente.

Per esempio, se la cella contiene @ 1, esso verrà sostituito dal valore della colonna corrente.

Anche se questa descrizione potrà apparire complicata, in realtà il modo di operare del programma è molto semplice: con un po' di pratica e sperimentazione diventerà molto facile costruire relazioni matematiche anche molto complesse.

Procediamo nel nostro inserimento di linee, digitando:

```
11 DEFINE@ 4= @ 1* @ 2
12 DEFINE E @ =A @ +B @ +C @ +D @
```

Anche queste istruzioni seguono le regole precedenti: la 11 significa "Poni in ogni cella della colonna 4 il risultato del prodotto tra ogni cella della colonna 1 e ogni cella della colonna 2".

La linea 12 significa: "Poni in ogni colonna della riga E il risultato della somma tra ogni riga A, B, C e D".

Nell'inserimento di ogni linea di programma si può naturalmente usare il tasto DEL per correggere eventuali errori di digitazione e se vi accorgete di

aver commesso un errore in una linea precedente, potete utilizzare il comando CHANGE per sostituirla in un secondo tempo. Se vi siete dimenticati una linea, usate INSERT per inserirla nella sua giusta posizione.

Si noti che l'ordine in cui vengono immesse le istruzioni DEFINE è molto importante, poiché ognuna di esse può alterare certi contenuti delle celle che verranno utilizzate dalle DEFINE seguenti.

Per fare un po' di pratica con il modello che abbiamo appena definito ritornate al modo di operazione normale, premendo il tasto E (EXIT): lo schermo si pulisce, visualizzando i nomi di righe e colonne, le quali per ora contengono degli zeri. Il prossimo passo sarà, come è ovvio, sostituire questi zeri con i valori che ci interessano.

L'inserimento dei dati

Il comando CHANGE VALUE permette di modificare qualsiasi valore sullo schermo. Per utilizzarlo premete il tasto C, e il programma vi chiederà riga e colonna. Usate la designazione vista precedentemente (per esempio A3, riga A e colonna 3). Vi verrà quindi richiesto il valore da inserire nella cella considerata.

Una volta digitato anche quest'ultimo, esso sarà visualizzato nella sua giusta posizione.

Il primo passo sarà inserire i valori dei Tassi nella colonna 2: usate CHANGE VALUE per inserire i valori elencati nello specchio precedente. Infine andranno inseriti i valori per il Capitale nella colonna 1, usando l'analogo procedimento.

Alla fine dell'inserimento dei dati sarà sufficiente premere il tasto R (REDRAW SCREEN) per avere visualizzati dopo un certo tempo i risultati di tutti i calcoli.

Avete così creato il vostro primo modello finanziario; fate qualche esperimento, modificando uno o più valori del Capitale o del Tasso, e premete ancora R: in pochi secondi vengono visualizzati i risultati per la nuova situazione modificata.

Una volta compreso appieno il funzionamento di SuperCalc potrete inserirvi modelli sempre più sofisticati ed effettuare in un baleno analisi "Cosa se?", che richiederebbero ore e ore di calcoli, se fatte con carta e matita.

Registrazione e caricamento dei modelli

SuperCalc è dotato delle funzioni SAVE WORKSHEET e LOAD WORKSHEET, che permettono di salvare e ricaricare su disco i modelli e i dati che sono stati inseriti.

Esse sono accessibili attraverso la pressione dei tasti S ed L, ed accettano nomi di file lunghi fino a 10 caratteri.

SuperCalc è fornito per essere utilizzato con il disk drive; se invece desiderate utilizzare il registratore a cassette, sostituite semplicemente all'interno del programma le seguenti linee:

```
510 OPEN1,1,1,I$
610 OPEN1,1,0,I$:IF ST<>0THEN670
1610 OPEN1,1,1,I$:FORX=1TO100
1710 OPEN1,1,0,I$:IF ST<>0THEN1750
```

Ecco ora un elenco di tutti i comandi e delle istruzioni disponibili all'interno di MultiCalc:

CHANGE VALUE: permette di cambiare il valore contenuto in una qualunque delle celle dello spreadsheet. Usare la designazione standard per la cella (esempio: A3)

REDRAW SCREEN: ricalcola tutti i valori sulla base delle istruzioni DEFINE inserite in modo programma.

SAVE WORKSHEET: salva il contenuto del foglio elettronico, compreso il set di istruzioni inserito in modo programma.

LOAD WORKSHEET: carica in memoria un foglio elettronico salvato con il comando di cui sopra.

EXIT: ritorna al BASIC.

PROGRAM MODE: entra in modo programma, in cui possono essere usati i seguenti comandi:

INSERT: inserisce una o più linee di programma nella lista di istruzioni.

LIST: esegue un listato di una parte di programma. Il listato viene effettuato a partire da un dato numero di linea fino a quel numero più venti.

DELETE: permette di cancellare un gruppo di linee di programma.

CHANGE: permette di modificare una linea di programma inserita in maniera errata.

SAVE: salva su memoria di massa le linee di programma, ma non i numeri del foglio elettronico.

READ: legge da memoria di massa le linee di un programma salvato con il comando precedente.

EXIT: ritorna al modo di operazione normale.

Le seguenti istruzioni sono invece utilizzabili in modo programma:

NAME: assegna il nome ad una riga o colonna, e definisce la larghezza di una colonna. Esempio: NAME C3=TEST,4.

DEFINE: pone una cella uguale a un'espressione. Esempio: DEFINE A3 = C3 + B1.

Note finali

SuperCalc è limitato da un numero massimo di 20 righe e, in dipendenza dalla larghezza selezionata, 5 o 6 colonne. Questa limitazione lo rende più che mai adatto all'elaborazione di calcoli di gestione casalinga, anche se è teoricamente possibile effettuare calcoli finanziari più complessi. Infine, ancora qual-

che parola sull'utilizzo del programma:
 - il comando DEFINE @@=0 azzerava completamente il foglio elettronico;
 - si ricordi che le costanti possono esse-

re usate nelle formule, come ad esempio lo zero nell'istruzione di cui sopra;
 - il comando READ può essere usato anche per caricare un programma salvato con SAVE WORKSHEET, senza

tuttavia caricare anche i dati numerici;
 - se si esce dal programma con EXIT, digitando GOTO100 è possibile rientrarvi senza perdere i dati correntemente in RAM.

SuperCalc per C16

```

10 REM SUPERCALC PER C16           :rem 120
20 REM                             :rem 70
30 DIMPC%(110),PC$(110),WK(20,9),RN$(20),
   CN$(9),CW(9),CP(9)             :rem 43
40 X=0:SL=0:LL=0:L=0:CC=0:CR=0    :rem 10
45 R$(1)="DEFINE":R$(2)="NAME":R$(3)="END
   ":PC%(1)=3                     :rem 210
50 S$="{ 39 SPAZI}"               :rem 93
60 C$="{HOME}{ 23 GIU'}"         :rem 232
65 COLOR4,1:COLOR0,1:PRINT"{WHT}"
                                   :rem 209
70 PRINTCHR$(14);"{CLR}"LEFT$(C$,8)TAB(10)
   )"SUPERCALC C16"               :rem 193
80 REM                             :rem 76
90 PRINTLEFT$(C$,24)TAB(8)"RETURN PER CON
   TINUARE{ 2 SPAZI}";:GOSUB10000
                                   :rem 216
100 GOSUB5000                     :rem 214
105 GOSUB6000                     :rem 220
110 PRINT"{RVS}P{OFF}ROGRAM MODE {RVS}C
   {OFF}HANGE VALUE {RVS}R{OFF}EDRAW SCR
   EEN"                             :rem 180
120 PRINT"{RVS}S{OFF}AVE WORKSHEET {RVS}
   L{OFF}OAD WORKSHEET {RVS}E{OFF}XIT"
                                   :rem 209
130 GOSUB10000                   :rem 5
160 IFX$="R"THEN100               :rem 53
170 IFX$="E"THENPRINT"{CLR}":END:rem 13
180 IFX$="S"THEN500               :rem 60
190 IFX$="L"THEN600               :rem 55
200 IFX$="P"THEN1000              :rem 94
210 IFX$="C"THEN400               :rem 37
220 GOTO105                       :rem 99
400 GOSUB6000:PRINT"CAMBIA RIGA
   { 3 SPAZI}";:GOSUB10000:PRINT
                                   :rem 140
410 CR=ASC(X$)-64:IFCR<1ORCR>20THEN105
                                   :rem 8
415 PRINT"CAMBIA COLONNA:";:GOSUB10000:PR
   INT                             :rem 249
420 CC=VAL(X$):IFCC=0THEN105      :rem 138
430 PRINT"NUOVO VALORE:";:I=20:GOSUB9000:
   WK(CR,CC)=VAL(I$)               :rem 52
440 GOSUB5700:GOTO105             :rem 237
500 GOSUB6000:PRINT"SAVE WORKSHEET:";:I=1
   0:GOSUB9000:IFI$=""THEN105      :rem 102
510 OPEN1,8,2,I$+",S,W"          :rem 197
515 FORX=1TO100:PRINT#1,PC%(X)CHR$(13)CHR
   $(34)PC$(X)CHR$(13);           :rem 228
520 IFPC%(X)<>3THENNEXT           :rem 162
530 FORCR=1TO20:FORCC=1TO9:PRINT#1,WK(CR,
   CC)CHR$(13);:NEXT:NEXT         :rem 220
540 PRINT#1,CHR$(13);:CLOSE1:GOTO100
                                   :rem 124
600 GOSUB6000:PRINT"LOAD WORKSHEET:";:I=1
   0:GOSUB9000:IFI$=""THEN105      :rem 88
610 OPEN1,8,2,I$+",S,R":IFST<>0THEN670
                                   :rem 167
620 FORX=1TO100:INPUT#1,X$,PC$(X):T=ST:PC
   %(X)=VAL(X$):IFT<>0THEN670      :rem 88
630 IFPC%(X)<>3THENNEXT           :rem 164
640 FORCR=1TO20:FORCC=1TO9:INPUT#1,X$:T=S
   T:WK(CR,CC)=VAL(X$):IFST<>0THEN670
                                   :rem 249
650 NEXT:NEXT:CLOSE1:RC=1:GOTO100
                                   :rem 117
670 PRINT"I/O{ 2 SPAZI}ERROR.":PRINT"UN T
   ASTO PER CONTINUARE ";:GOSUB10000
                                   :rem 235
680 CLOSE1:RC=1:GOTO 100          :rem 134
1000 FL=1                          :rem 193
1010 PRINT"{HOME}";:FORX=FLTOFL+20:PRINTS
   $:PRINT"{SU}"X;R$(PC%(X))"PC$(X):N
   EXT                             :rem 157
1020 GOSUB6000                    :rem 9
1030 PRINT"{RVS}I{OFF}NSERT {RVS}L{OFF}IS
   T {RVS}D{OFF}ELETE {RVS}C{OFF}HANGE
   {RVS}S{OFF}AVE {RVS}R{OFF}EAD"
                                   :rem 34
1035 PRINT"{RVS}E{OFF}XIT"        :rem 248
1040 GOSUB10000                   :rem 54
1050 IFX$="E"THENRC=1:GOTO100     :rem 205
1060 IFX$="I"THEN1200             :rem 142
1070 IFX$="L"THEN1300             :rem 147
1080 IFX$="D"THEN1400             :rem 141
1090 IFX$="C"THEN1500             :rem 142
1100 IFX$="S"THEN1600             :rem 151
1110 IFX$="R"THEN1700             :rem 152
1120 GOTO1020                     :rem 192
1200 GOSUB6000:PRINT"INSERIRE PRIMA DELLA
   LINEA:";:GOSUB8990:SL=VAL(I$)
                                   :rem 121
1210 IFSL=0THEN1020               :rem 81
1220 PRINT"QUANTE LINEE?:";:GOSUB8990:N=V
   AL(I$):N1=100-N:IFN=0THEN1020
                                   :rem 247
1230 FORX=N1TOSLSTEP-1:PC%(X+N)=PC%(X):PC
   $(X+N)=PC$(X):NEXT             :rem 120
1235 N=N-1:FORL=SLTOSL+N:GOSUB8000:NEXT
                                   :rem 1
1240 GOTO1010                     :rem 194
1300 GOSUB6000:PRINT"INIZIO ALLA LINEA:";
   :GOSUB8990                       :rem 115
1320 IFVAL(I$)=0ORVAL(I$)>80THEN1020
                                   :rem 61
1330 FL=VAL(I$):GOTO1010          :rem 108
1400 GOSUB6000:PRINT"ELIMINA DALLA LINEA:
   ";:GOSUB8990:SL=VAL(I$):IFSL=0THEN10
   20                               :rem 99
1410 PRINT"ALLA:";:GOSUB8990:LL=VAL(I$):I
   FLL=0THEN1020                   :rem 42
1420 N=LL-SL+1:FORX=SLTO100-N:PC%(X)=PC%(
   X+N):PC$(X)=PC$(X+N):NEXT      :rem 241
1430 GOTO1010                     :rem 195
1500 GOSUB6000:PRINT"CAMBIA LA LINEA:";:G
   OSUB8990:L=VAL(I$)               :rem 23
1510 GOSUB8000:GOTO1010           :rem 68
1600 GOSUB6000:PRINT"SAVE PROGRAM:";:I=10
   :GOSUB9000:IFI$=""THEN1020      :rem 33
1610 OPEN1,8,2,I$+",S,W":FORX=1TO100
                                   :rem 18
1620 PRINT#1,PC%(X)CHR$(13)CHR$(34)PC$(X)
   CHR$(13);                       :rem 247
1630 IFPC%(X)<>3THENNEXT           :rem 213
1640 CLOSE1:GOTO1000              :rem 166
1700 GOSUB6000:PRINT"LOAD PROGRAM:";:I=10
  
```

```

:GOSUB9000:IFI$=""THEN1020 :rem 19
1710 OPEN1,8,2,I$+",S,R":IFST<>0THEN1750 :rem 9
1720 FORX=1TO100:INPUT#1,X$,PC$(X):PC%(X)
=VAL(X$) :rem 133
1730 IFPC$(X)<>3THENNEXT :rem 214
1740 CLOSE1:GOTO1000 :rem 167
1750 PRINT"I/O{ 2 SPAZI}ERROR":PRINT"UN T
ASTO PER CONTINUARE ";:GOSUB10000 :rem 237
1760 CLOSE1:GOTO1000 :rem 169
4999 END :rem 182
5000 IFRC=0THEN5400 :rem 78
5005 FORX=1TO9:RN$(X)="" :CN$(X)="" :CW(X)=
0:CP(X)=0:NEXT :rem 218
5007 FORX=10TO20:RN$(X)="" :NEXT :rem 79
5010 RC=0:FORL=1TO100:IFPC%(L)=3THEN5310 :rem 93
5020 IFPC%(L)<>2THEN5300 :rem 78
5030 X$=LEFT$(PC$(L),1):N$=MID$(PC$(L),2,
1) :rem 25
5040 IFX$="C"THEN5100 :rem 141
5050 IFX$="R"THEN5200 :rem 158
5060 GOTO5900 :rem 210
5100 FORX=4TOLEN(PC$(L)):IFMID$(PC$(L),X,
1)<>"",THENNEXT:LN=5:GOTO5120 :rem 195
5110 LN=VAL(MID$(PC$(L),X+1)) :rem 85
5120 N=VAL(N$):IFN=0THEN5900 :rem 122
5130 CN$(N)=MID$(PC$(L),4,X-4):CW(N)=LN :rem 140
5140 GOTO5300 :rem 203
5200 N=ASC(N$)-64:IFN<1ORN>20THEN5900 :rem 147
5210 RN$(N)=MID$(PC$(L),4) :rem 107
5300 NEXTL :rem 83
5310 CP(1)=0:FORX=2TO9:CP(X)=CP(X-1)+CW(X
-1)+ABS(CW(X)>0):NEXT :rem 164
5400 PRINT"{CLR}"C$"ATTENDERE..." :rem 126
5405 PRINT"{HOME}{RVS}";:FORX=1TO9:PRINTT
AB(CP(X)+7)RIGHT$(S$+CN$(X),CW(X)); :rem 236
5410 NEXT:PRINT :rem 208
5420 FORX=1TO20:PRINT"{RVS}"LEFT$(RN$(X),
6):NEXT :rem 130
5450 FORL=1TO100:IFPC%(L)=3THEN5700 :rem 44
5460 IFPC%(L)<>1THEN5690 :rem 97
5470 X$=LEFT$(PC$(L),2):AC=VAL(RIGHT$(X$,
1)):IFAC<1ORAC>9THENAC=0 :rem 76
5480 AR=ASC(X$)-64:IFAR<1ORAR>20THENAR=0 :rem 168
5490 CC=AC:IFAC=0THENRC=1:FORCC=1TO9 :rem 246
5495 CR=AR:IFAR=0THENRR=1:FORCR=1TO20 :rem 111
5500 IFMID$(PC$(L),3,1)<>"="THEN5900 :rem 176
5510 X$=MID$(PC$(L),4):OP$="+":S=1:T=0 :rem 38
5520 FORX=STOLEN(X$):N$=MID$(X$,X,1):IFN$
="+ORN$="-ORN$="/ORN$="*ORN$="THEN5550 :rem 166
5540 NEXTX :rem 101
5550 N$=MID$(X$,S,X-S):S=X+1 :rem 74
5560 IFN$="0"ORVAL(N$)<>0THENV=VAL(N$):GO
TO5610 :rem 26
5565 IFLEN(N$)<>2THEN5900 :rem 175
5570 TR$=LEFT$(N$,1):TC$=RIGHT$(N$,1) :rem 59
5580 TR=ASC(TR$)-64:IFTR<1ORTR>20THENTR=C
R :rem 168
5590 TC=VAL(TC$):IFTC<1ORTC>9THENTC=CC :rem 155
5600 V=WK(TR,TC) :rem 186
5610 IFOP$="+THENT=T+V :rem 95
5620 IFOP$="-THENT=T-V :rem 100
5630 IFOP$="/THENT=T/V :rem 105
5640 IFOP$="*THENT=T*V :rem 96
5650 IFX<LEN(X$)THENOP$=MID$(X$,X,1):GOTO
5520 :rem 185
5660 WK(CR,CC)=T :rem 156
5670 IFRR=1THENNEXTCR :rem 118
5680 IFRC=1THENNEXTCC :rem 89
5690 RR=0:RC=0:NEXTL :rem 230
5700 PRINT"{HOME}{GIU'}";:FORCR=1TO20:IFR
N$(CR)=""THEN5795 :rem 34
5710 FORCC=1TO9:PRINTTAB(CP(CC)+7); :rem 122
5720 X$=MID$(STR$(WK(CR,CC)),2) :rem 220
5730 IFLEN(X$)<=CW(CC)THEN5790 :rem 248
5740 FORX=1TOLEN(X$):IFMID$(X$,X,1)<>".T
HENNEXT:GOTO5790 :rem 227
5750 N$=LEFT$(X$,X-1) :rem 126
5760 IFLEN(N$)<CW(CC)THENN$=N$+MID$(X$,X,
CW(CC)-LEN(N$)) :rem 230
5780 X$=N$ :rem 255
5790 PRINTRIGHT$(S$+X$,CW(CC));:NEXTCC :rem 74
5795 PRINT:NEXTCR :rem 117
5800 RETURN :rem 173
5900 GOSUB6000:PRINT"ERRORE DI SINTASSI N
ELLA"L :rem 161
5910 PRINT"UN TASTO PER CONTINUARE ";:GOS
UB10000 :rem 147
5920 RETURN :rem 176
6000 PRINTLEFT$(C$,22)S$:PRINTS$:PRINTS$:
PRINTLEFT$(C$,22); :rem 118
6010 RETURN :rem 167
8000 GOSUB6000:PRINTL; :rem 92
8010 I=35:GOSUB9000:IFI$=""THEN8000 :rem 232
8020 FORX=1TOLEN(I$):IFMID$(I$,X,1)<>" "T
HENNEXT :rem 73
8030 PC$(L)=MID$(I$,X+1):PC%(L)=0 :rem 244
8040 X$=LEFT$(I$,X-1):FORX=1TO3:IFX$=R$(X
)THENPC%(L)=X :rem 235
8050 NEXT:RETURN :rem 38
8990 I=3:GOSUB9000 :rem 22
8992 IFVAL(I$)<1ORVAL(I$)>100THENI$="" :rem 167
8995 RETURN :rem 191
8999 I=3 :rem 156
9000 I$="" :rem 183
9010 GETX$:IFX$=""THEN9010 :rem 229
9020 C=ASC(X$):IFC=20THEN9060 :rem 152
9025 IFC=13THENPRINT" ":RETURN :rem 93
9030 IF(C>31ANDC<95)OR(C>192ANDC<219)THEN
9040 :rem 74
9035 GOTO9010 :rem 212
9040 IFLEN(I$)=ITHEN9010 :rem 120
9050 PRINTX$;:I$=I$+X$:GOTO9010 :rem 71
9060 IFI$=""THEN9010 :rem 69
9070 PRINTX$;:I$=LEFT$(I$,LEN(I$)-1):GOTO
9010 :rem 105
10000 REM :rem 213
10010 GETX$:IFX$=""THEN10010 :rem 53
10020 PRINTX$;:RETURN :rem 81

```



ZAP! per C16-Plus/4

di **T. Marazita**
trad. e adatt.
di **A. Borra**

Chissà che effetto farebbe avere la coda! Certo, detta così la cosa appare banale e priva di senso pratico, ma proviamo ad astrarci un attimo dalla realtà e lasciamo correre la mente. Sicuramente sarebbe di impiccio: rischieremo di chiuderla dentro le porte, o di farcela calpestare in autobus. E poi come si potrebbe dormire supini, con quella protuberanza che ci spunta dal didietro?

D'altro canto ci sono anche i lati positivi: aumenterebbe il nostro equilibrio (e che fine farebbero gli equilibristi del circo?), potrebbe magari servire come terzo arto (chissà come sarebbe una partita di calcio!?) e certamente diverrebbe un importante simbolo estetico. Ma sicuramente il più grosso problema sarebbe quello della locomozione; se poi addirittura la coda continuasse a crescere con continuità... eh no! calma: questa non è più fantasia: è ZAP!, il divertente gioco che stiamo per presentare.

Il gioco

Come si sarà capito, il gioco simula il movimento, in un ambiente chiuso, di un serpente dalla coda che si allunga sempre più. Lo scopo del gioco è quello di guidarlo alla caccia dei bersagli che di volta in volta appaiono sullo schermo, e velocemente, perché dopo poco essi scompaiono. Il movimento è ottenuto mediante la tastiera, con gli otto tasti attorno alla "L" per la scelta delle otto direzioni possibili (ma l'esperienza consiglia di non utilizzare le quattro direzioni diagonali, perché difficili da controllare), mentre il tasto "L" consente di "congelare" il gioco per permettere una boccata di respiro nei momenti di maggiore difficoltà: ciò è soprattutto utile quando la coda è così lunga da intralciare pesantemente i movimenti (e vedrete che, data la velocità del gioco, ciò avverrà assai presto).



Il gioco avrà termine dopo 6 "incidenti", che possono essere sia scontri con la propria coda che contatti con i muri elettrici (zap!) che delimitano la zona di movimento. Alla fine viene segnalato il punteggio e la percentuale di bersagli colpiti.

Dopo questo gioco vi convincerete che "Mamma Natura" non ha sbagliato a toglierci la coda!

Commenti al listato

Dopo la schermata introduttiva e le

istruzioni (i due GOSUB alle linee 150 e 170) inizia il gioco vero e proprio, con il disegno dei limiti della zona di movimento (linee 230-260). Quindi c'è il ciclo di decodifica del tasto premuto e la chiamata alle varie routine: aggiornamento punteggio, movimento serpente e posizionamento bersagli. Al raggiungimento di 6 scontri il controllo passa alla routine di stampa dei risultati finali (870-970) che propone una ulteriore partita.

Zap! per C16-Plus/4

```

100 REM**ZAP!** :rem 41
110 COLOR4,10,1:COLOR0,1:PRINT"{CLR}" :rem 19
120 PRINT"{WHT}" :rem 105
130 DIMWM(255) :rem 255
140 EE=18:HP=EE-1 :rem 188
150 GOSUB1280 :rem 225
160 DIMA(255) :rem 159
170 GOSUB1080 :rem 225
180 COLOR0,2 :rem 166
190 PRINT"{CLR}" :rem 254
200 COLOR0,1 :rem 158
210 PO=4070 :rem 58
220 S=1 :rem 85
230 FORX=3113TO3112+38:POKEX,100:NEXT :rem 208
240 FORX=3112+40TO3072+(40*23)STEP40:POKE :rem 224
X,103:NEXT
250 FORX=3073+(40*24)TO4070:POKEX,99:NEXT :rem 150
260 FORX=4071-40TO4071-(40*22)STEP-40:POK :rem 21
EX,101:NEXT
270 GOSUB1060 :rem 224
280 FORI=1TO10:GETA$:NEXT :rem 52
290 GOTO550 :rem 110
300 A$="O":GOTO320 :rem 208
310 GETA$ :rem 217
320 IFA$="K"THENDIR=-1 :rem 254
330 IFA$="O"THENDIR=-40 :rem 54
340 IFA$=":"THENDIR=1 :rem 194
350 IFA$="."THENDIR=40 :rem 234
360 IFA$="I"THENDIR=-41 :rem 52
370 IFA$="P"THENDIR=-39 :rem 67
380 IFA$="/"THENDIR=41 :rem 239
390 IFA$=","THENDIR=39 :rem 244
400 IFA$="L"THEN1020 :rem 71
410 PO=PO+DIR :rem 26
420 IFPO=4071ORPO=4032ORPO=3112ORPO=3151T :rem 234
HEN760:REM**ANGOLI**
430 IFPEEK(PO)=102THENGOSUB690 :rem 89
440 WM(HP)=PO:HP=HP+1 :rem 4
450 IFHP>EETHENHP=0 :rem 188
460 IFPEEK(PO)>98ANDPEEK(PO)<102THEN760 :rem 112
470 IFPEEK(PO)>102ANDPEEK(PO)<104THEN760 :rem 149
480 IFPEEK(PO)=81THEN760 :rem 178
490 POKEPO,81 :rem 0
500 R=INT(RND(1)*15)+1:IFS>5THENZZ=20 :rem 195
510 P1=INT(RND(1)*ZZ)+1:ZZ=50 :rem 192
520 IFR=3THEN550 :rem 177
530 IFP1=3THEN630 :rem 224
540 GOSUB980:GOTO310 :rem 193
550 IFS>6THEN540 :rem 184
560 Q=INT(RND(0)*1000)+(3072+160) :rem 191
570 IFQ>4030THEN560 :rem 75
580 IFPEEK(Q)<>32THEN560 :rem 156
590 S=S+1:H=H+1:TC=TC+1 :rem 65
600 POKEQ,102 :rem 213
610 A(H)=Q:IFCV=1THEN540 :rem 151
620 CV=1:GOTO300 :rem 165
630 REM**ELIMINA I BERSAGLI** :rem 182
640 P=INT(RND(0)*H)+1 :rem 150
650 IFPEEK(A(P))<>102THEN680 :rem 92
660 POKEA(P),32 :rem 62
670 S=S-1 :rem 222
680 A(P)=A(H):H=H-1:GOTO540 :rem 8
690 REM**COLPITO UN BERSAGLIO** :rem 128
700 SC=SC+10 :rem 140
710 S=S-1 :rem 217
720 EE=EE+2 :rem 71
730 IFEE>250THENEE=250 :rem 21
740 GOSUB1060 :rem 226
750 RETURN :rem 124
760 CR=CR+1:REM**URTO** :rem 112
770 GOSUB1060 :rem 229
780 IFCS>LTHENL=CS :rem 156
790 CS=0 :rem 163
800 EE=18:HP=17 :rem 63
810 TP=0 :rem 170
820 CV=0 :rem 160
830 FORT=1TO8:FORX=1TO100:NEXT:POKEPO,209 :rem 93
:FORX1=1TO100:NEXT:POKEPO,81:NEXT
840 FORX=0TO255:WM(X)=0:NEXT:HP=17:TP=0 :rem 182
850 IFCR=6THEN870 :rem 2
860 GOTO180 :rem 112
870 PRINT"{CLR}{" 6 GIU'"}" :rem 105
880 PRINT"HAI FATTO "SC" PUNTI" :rem 43
890 PRINT :rem 46
900 PRINT :rem 38
910 PRINT"IL PUNTEGGIO MASSIMO ERA : "TC*1 :rem 255
0
920 PRINT :rem 40
930 PRINT"HAI PRESO IL ";INT((SC/(TC*10)) :rem 215
*100);"% DEI BERSAGLI"
940 POKE239,0 :rem 198
950 INPUT"{ 2 GIU'}VUOI GIOCARE ANCORA";P :rem 128
L$
960 IFLEFT$(P$,1)="S"THENRUN :rem 227
970 PRINT"{ 3 GIU'}CIAO!":END :rem 242
980 LOC=WM(TP):TP=TP+1:REM**CANCELLA LA C :rem 13
ODA**
990 IFTP>EETHENTP=0 :rem 221
1000 POKELOC,32 :rem 95
1010 RETURN :rem 162
1020 FORE=1TO171:GETA$:IFA$<>" "THEN320 :rem 200
1030 NEXTE:SC=SC-10:IFSC<0THENSC=0 :rem 118
1040 GOSUB1060 :rem 12
1050 GOTO1020 :rem 194
1060 PRINT"{HOME}{" 3 DES}PUNTI="SC;TAB(18 :rem 227
);"NO. DI CRASHES="CR
1070 RETURN :rem 168
1080 INPUT"{ 2 GIU'}{YEL}VUOI LE ISTRUZIO :rem 242
NI";B$
1090 IFLEFT$(B$,1)="S"THEN1110 :rem 130
1100 RETURN :rem 162
1110 PRINT"{CLR}{" 2 GIU'}DEVI CONTROLLARE :rem 182
UNO STRANO MISSILE!"
1120 PRINT"{GIU'}QUANDO I BERSAGLI '[<+>] :rem 218
' APPAIONO SULLO"
1130 PRINT"SCHERMO, DEVI PRENDERLI. PER M :rem 216
UOVERE"
1140 PRINT"IL MISSILE USA LA CROCE:" :rem 94
1150 PRINT"{GIU'}{" 13 SPAZI}I{" 4 SPAZI}O :rem 145
{" 4 SPAZI}P"
1160 PRINT"{ 18 SPAZI}↑" :rem 247
1170 PRINT"{GIU'}{" 13 SPAZI}K <- L -> :rem 80
1180 PRINT"{GIU'}{" 18 SPAZI}I" :rem 245
1190 PRINT"{ 13 SPAZI},{ 4 SPAZI}. :rem 37
{" 4 SPAZI}"/"
1200 PRINT"{GIU'}E LE SUE DIAGONALI. :rem 79
{" 2 SPAZI}CON 'L' SI PUO'"

```

LOAD

ZAP PER C16

```

1210 PRINT"FERMARE IL GIOCO." :rem 203
1220 PRINT"OGNI BERSAGLIO VALE 10 PUNTI,
      MA IL" :rem 195
1230 PRINT"MISSILE SI ALLUNGA..."
      :rem 215
1240 PRINT"{GIU'}IL GIOCO FINISCE DOPO 6
      CRASHES!" :rem 66
1250 PRINT"{ 2 GIU'}PREMI 'SPAZIO' PER IN
      IZIARE IL GIOCO"; :rem 223
1260 GETZZ$:IFZZ$=""THENGOTO1260:rem 212
1270 RETURN :rem 170
1280 PRINT"{ 3 GIU'}{ 8 SPAZI}ZZZZZZ
      { 3 SPAZI}AAA{ 3 SPAZI}PPPPPP
      { 2 SPAZI}Q" :rem 95
1290 PRINT"{ 13 SPAZI}Z{ 2 SPAZI}A
      { 3 SPAZI}A{ 2 SPAZI}P{ 5 SPAZI}P Q
      " :rem 234
1300 PRINT"{ 12 SPAZI}Z{ 2 SPAZI}A
      { 5 SPAZI}A P{ 5 SPAZI}P Q":rem 226
1310 PRINT"{ 10 SPAZI}ZZ{ 3 SPAZI}AAAAAAA
      PPPPPP{ 2 SPAZI}Q" :rem 194
1320 PRINT"{ 9 SPAZI}Z{ 5 SPAZI}A
      { 5 SPAZI}A P{ 7 SPAZI}Q" :rem 148
1330 PRINT"{ 8 SPAZI}Z{ 6 SPAZI}A
      { 5 SPAZI}A P" :rem 196
1340 PRINT"{ 8 SPAZI}ZZZZZZ A{ 5 SPAZI}A
      P{ 7 SPAZI}Q" :rem 88
1350 PRINT"{ 3 GIU'}{YEL}{ 8 SPAZI}CURSOR
      #2{ 3 SPAZI}AGOSTO 1978" :rem 68
1360 PRINT"{GIU'}{ 5 SPAZI}COPYRIGHT 1978
      - TOM MARAZITA" :rem 180
1370 PRINT"{ 3 GIU'}{PUR}{ 4 SPAZI}ADATTA
      MENTO SU C-16 DI A. BORRA" :rem 142
1380 RETURN :rem 172
  
```



Novità firmate Jackson.



CON CASSETTA

Michel Benelfoul
GIOCHI, TRUCCHI E CONTI

Questo libro è destinato soprattutto ai programmatori principianti: con i suoi 29 programmi presentati con una ricca documentazione e pronti per essere eseguiti. Gli argomenti sono tanti e rispondono alle esigenze più disparate. Il BASIC è quello dei calcolatori PET/CBM di cui vengono sfruttate a pieno le molte funzioni.

Cod. 553D Pag. 164 Lire. 24.000

Rita Bonelli
Luciano Pazzuconi - Fabio Racchi
Giovanni Valerio
**COMMODORE 64
LA GRAFICA E IL SUONO**

Ogni argomento viene spiegato e accompagnato da numerosi esempi commentati. Nel libro sono listati moltissimi programmi; essi sono registrati sulla cassetta allegata. Inoltre la cassetta contiene anche altri programmi, ai quali si fa riferimento nel testo, ma senza listarli.
Cod. 409B Pag. 270 Lire. 34.000



**GRUPPO
EDITORIALE
JACKSON**

ritagliare (o fotocopiare) e spedire in busta chiusa a:

GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

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

Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca

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

N° _____

Allego fotocopia di versamento su vaglia postale a voi intestato

Nome _____

Cognome _____

Via _____

Cap _____

Città _____

Prov. _____

Data _____

Firma _____

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

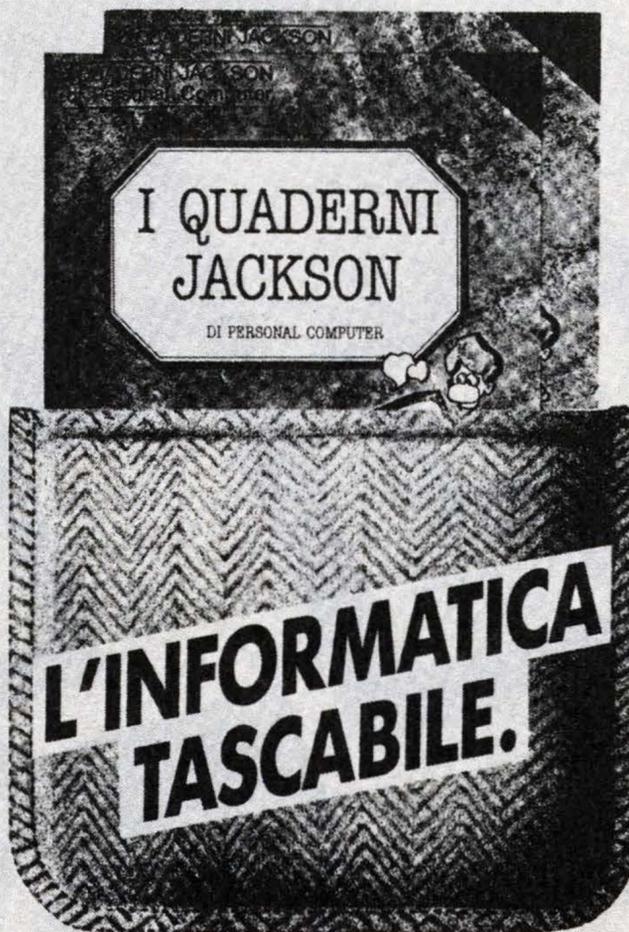
ORDINE
MINIMO
L. 50.000

Partita I.V.A. _____

LA TUA BIBLIOTECA PRATICA D'INFORMATICA PERSONALE.

I Quaderni Jackson, tanti volumi monografici per conoscere bene il personal computer e l'informatica. Nei Quaderni Jackson c'è tutto quello che è importante sapere sui computer la programmazione, i linguaggi, il software, le applicazioni e i nuovi sviluppi dell'informatica.

Quaderni Jackson: l'informatica a tutti i livelli, in una collana aperta, pratica, essenziale, aggiornata. L'informatica tascabile per chi vuole saperne di più e compiere così un salto di qualità nel mondo di oggi e di domani. Ogni settimana, 1 volume.



1
Vivere col personal computer

2
Dentro e fuori la scatola
hardware e periferiche di un personal computer.

3
Ed è subito BASIC, vol 1

4
A ciascuno il suo Personal come sceglierlo in base alle esigenze

5
Ed è subito BASIC, vol 2

6
To do or not to do: come aver cura del proprio personal

7
Strutturare il software (con elementi di linguaggio Pascal)

8
Dizionario informatico

9
Stendere un programma come si deve: algoritmi e cose serie

10
La linfa del computer software di base e sistemi operativi

11
Il sistema operativo CP/M, il "software bus"

12
Sistema Operativo MS DOS, lo standard creato dalla IBM

13
I package applicativi: merceologia del software prêt a porter

14
Scrivere un gioco di avventura sul personal computer

15
Pronto, VisiCalc e gli onesti spreadsheet

16
Il BASIC per l'immagine le basi della computergrafica

17
Il word processing parole, parole: ma elaborate col personal

18
Lotus 1-2-3 e Symphony il fascino discreto dell'integrazione

19
Dimensione MSX

20
Dacci oggi i nostri dati: in BASIC e Pascal vol. I

21
Dacci oggi i nostri dati: in BASIC e Pascal vol. II

22
Il dBase, il principe dei data base

23
A qualcuno piace FORTH! anatomia d'un linguaggio astuto

24
Disegnare col personal computer

25
I famosi Bitest prove rivisitate dei personal più popolari

26
Multiplan, spreadsheet multistrato

27
FORTRAN e COBOL linguaggi sempreverdi

28
I famosi Softest prove rivisitate dei package più popolari

29
Suono e Musica col Personal Computer

30
Calcolo numerico in BASIC

Ogni settimana a sole lire 6000 in edicola

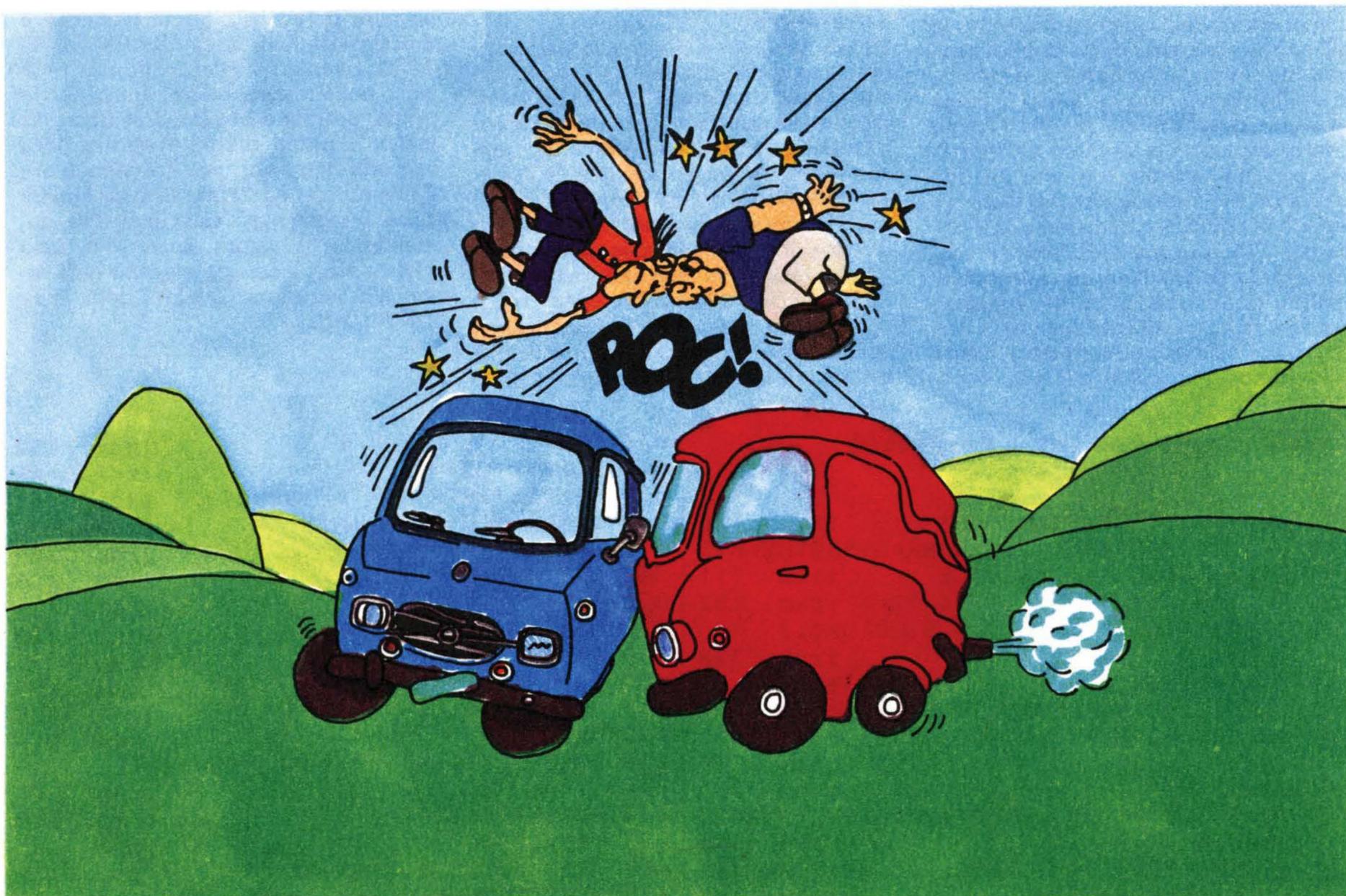


GRUPPO EDITORIALE **JACKSON**
DIVISIONE GRANDI OPERE



Guida pericolosa per C16-Plus/4

di A. Borra



Il gioco, semplice ed essenziale, prende spunto da uno presente nelle "vecchie" sale giochi, quando ancora non era scoppiato il "boom" dei videogame: si trattava di un grosso cubo, trasparente in alto, al quale erano uniti un volante e due pedali: lo scopo era quello di pilotare dentro il cubo una pallina di metallo lungo un piano inclinabile nelle quattro direzioni, in modo che riuscisse ad evitare le buche che si trovano lungo il percorso per cadere infine in quella finale. Due erano le grandi attrattive di quel banale gioco: primo il fatto

che la pallina si trovava in un labirinto a forma di strada, con tanto di automobili e pedoni. Sembrava proprio di essere "grandi" alla guida della pallina: "Papà, guido proprio come te!". La seconda era che la pallina che riusciva ad arrivare alla fine veniva trasformata... in una gomma da masticare: era questo un premio che dava una gioia assai maggiore della semplice patente di guida!

Il tempo è passato, ma è rimasta però la memoria di quel vecchio gioco ormai introvabile: ora il C16, insieme ai

ricordi, è riuscito a far rivivere quei momenti, grazie a "Guida pericolosa", il gioco che qui vi presentiamo. Peccato che non sia stato possibile trasformare la pallina vincente in un chewing-gum: che sia questo uno di quei famosi limiti fisici del computer di cui tanto si sente parlare oggi?

Il gioco

Veniamo finalmente al gioco vero e proprio, come è stato realizzato sul C16: esso consiste nel guidare una pallina rossa (inizialmente posta in alto a sinistra) lungo un percorso labirintico, in modo che raggiunga (in basso a destra) il buco blu, evitando però tutti quelli

neri che intralciano il percorso. La pallina può essere guidata solo mediante l'inclinazione del suo piano di movimento: due indicatori permettono di determinare, istante per istante, la posizione del piano, in modo da avere il massimo controllo sulla pallina. Il movimento è controllato da quattro tasti:

"Z" per inclinare il piano verso sinistra
 "X" per inclinare il piano verso destra
 "/" per abbassare la parte superiore del piano (e quindi alzare quella inferiore)

"," per il movimento inverso.

Naturalmente, non si vedrà realmente inclinarsi il piano di movimento della pallina, ma essa incomincerà a muoversi nella direzione data dalle frecce degli indicatori, con velocità proporzionale all'inclinazione indicata. Il gioco tiene conto anche dell'azione dell'attrito: se gli indicatori mostrano una duplice inclinazione, esempio in basso e a destra, la pallina non andrà né verso il basso né verso destra, e ciò perché l'attrito contro le pareti del labirinto ne impedisce ogni movimento: solo uno dei due indicatori deve essere lontano dalla posizione di equilibrio; questo rende il gioco più difficile e appassionante: il controllo della pallina è veramente problematico. Una particolarità: per facilitare il rallentamento della pallina, che in certi casi (ve ne accorgete!) deve essere immediato, la pressione del tasto oppo-

sto a quello di inclinazione (esempio la pressione di "X" quando l'indicatore orizzontale è a sinistra) permette l'immediato ottenimento dell'equilibrio, quindi attenti a non farvi prendere dal panico!

Il cadere dentro uno dei buchi avrà l'effetto di dover ricominciare il gioco dall'inizio, ed è dura ripartire daccapo quando si era a un passo dalla meta! Al termine viene segnalato il tempo impiegato per effettuare il percorso e viene offerta la possibilità di giocare nuovamente: come si fa a rifiutare? E poi, quando avrete acquisito un po' di esperienza lanciate la grande sfida: "Papà, vieni che ti faccio vedere come si guida!!!"

Le caratteristiche del programma

Finora abbiamo parlato solo del gioco in sé, ma quello che conta è il modo con cui è stato realizzato e la descrizione delle scelte adottate, in modo da capire il funzionamento del programma per un suo eventuale miglioramento. Si osserva innanzitutto (linee 460-470) che il programma sfrutta gli attributi, cioè non disegna di volta in volta la pallina nella mappa di schermo, ma (dopo aver riempito di palline lo schermo: linea 160, il codice della pallina è 81) cambia solo il colore della casella (in mappa colori) nella quale si trova: tutte le altre palline

hanno il colore dello sfondo e quindi non sono visibili. In questo modo si sono risparmiate molte istruzioni: per posizionare la pallina è sufficiente cambiarne il colore invece di scriverla sullo schermo e poi colorarla.

Le variabili AX e AY tengono conto delle inclinazioni del piano, e il loro valore determina anche la velocità della pallina (linee 560-580). In caso di vittoria viene suonato un breve motivetto (linee 800-810, "DATA" in 1090). Il labirinto, infine è codificato nelle linee "DATA" 860-1060: gli "zero" indicano zona vuota, mentre gli "uno" indicano la presenza di un muro. Quindi è facile modificare la forma del labirinto con un'altra, ricordando che la sua dimensione è 21 righe per 37 colonne. In questo caso bisogna però modificare anche le palline nere (i "buchi"), la cui posizione è determinata dalle linee "DATA" 1070-1080: i numeri si riferiscono alla posizione nella mappa colore: quindi 0 è la prima posizione in alto a sinistra, 40 indica la prima casella a sinistra della seconda riga e così via. Per aumentare il numero dei buchi, e quindi della difficoltà, bisogna agire sulla linea 390, aumentando il valore finale del ciclo FOR... NEXT ed aggiungendo altri valori alle istruzioni "DATA" delle linee 1070-1080.

Guida pericolosa per C16-Plus/4

```

100 REM**GUIDA PERICOLOSA**           :rem 120
110 G=3072:H=2048:A=G:B=H:HX=H+939:HY=H+5
    19:W=160:C=3+16*6:VOL8             :rem 111
120 X=1:Y=1:NX=1:NY=1:S=54272        :rem 197
130 PRINT"{CLR}":COLOR4,4,6:COLOR0,2
                                         :rem 238
140 FORJ=1 TO37                        :rem 65
150 GJ=G+J:HJ=H+J:FORK=1 TO21         :rem 198
160 POKEHJ+40*K,113:POKEGJ+40*K,81
                                         :rem 176
170 NEXT                               :rem 215
180 POKEHJ,C:POKEGJ,W:POKEHJ+880,C:POKEGJ
    +880,W                               :rem 195
190 POKEHJ+920,113:POKEGJ+920,30:POKEGJ+9
    60,W:POKEHJ+960,C                   :rem 172
200 NEXT                               :rem 209
210 FORK=0 TO960 STEP40                :rem 20
220 POKEH+K,C:POKEG+K,W:POKEH+K+38,C:POKE
    G+K+38,W:POKEH+K+39,113:POKEG+K+39,31
                                         :rem 208
230 NEXT                               :rem 212
240 FORJ=920 TO928:POKEH+J,C:POKEG+J,W:NEX
    T                                     :rem 155
250 FORJ=950 TO959:POKEH+J,C:POKEG+J,W:NEX
    T                                     :rem 163
260 POKEH+39,C:POKEG+39,W:POKEH+79,C:POKE
    G+79,W:POKEH+999,C:POKEG+999,W
                                         :rem 103
270 POKEHY,4+16*6:POKEG+939,42        :rem 205
280 POKEHX,4+16*6:POKEG+519,42        :rem 199
290 FORJ=0 TO20                        :rem 62
300 READA$                             :rem 20
310 N=41                               :rem 132
320 FORK=1 TO37                        :rem 66
330 IFMID$(A$,K,1)="0" THEN350         :rem 37
340 POKEB+N,C:POKEA+N,W               :rem 150
350 N=N+1                              :rem 205
360 NEXT                               :rem 216
370 A=A+40:B=B+40                     :rem 114
380 NEXT                               :rem 218
390 FORJ=1 TO28                        :rem 72
400 READN                              :rem 254
410 POKEH+N,0                          :rem 225
420 NEXT                               :rem 213
430 POKEH+877,6+16*4:POKEG+877,81
                                         :rem 71
440 POKEH+41,2+16*4                    :rem 237
450 TI$="000000"                       :rem 251
460 IFAX<>0 THENPOKEHX+AX,113          :rem 239
470 IFAY<>0 THENPOKEHY+AY*40,113      :rem 129
480 GETA$:IFA$="X" ANDAX<0 ORA$="Z" ANDAX>0 T
    HENAX=0                              :rem 176
490 IFA$="X" ANDAX<10 THENAX=AX+1      :rem 109
500 IFA$="Z" ANDAX>-10 THENAX=AX-1     :rem 152
510 IFA$="/" ANDAY<0 ORA$=";" ANDAY>0 THENAY=
    0                                      :rem 230
520 IFA$="/" ANDAY<10 THENAY=AY+1      :rem 65
530 IFA$=";" ANDAY>-10 THENAY=AY-1     :rem 127
540 IFAX<>0 THENPOKEHX+AX,4+16*4       :rem 125
550 IFAY<>0 THENPOKEHY+AY*40,4+16*4
                                         :rem 15
560 VX=VX+AX*2:VY=VY+AY*2             :rem 74
570 IFABS(VX)>10 THENNX=X+1*SGN(VX):VX=0
                                         :rem 203
580 IFABS(VY)>10 THENNY=Y+1*SGN(VY):VY=0
                                         :rem 209
590 CX=PEEK(H+NX+Y*40) AND7           :rem 33

```

```

600 CY=PEEK (H+X+NY*40) AND7      :rem 26
610 CZ=PEEK (H+NX+NY*40) AND7      :rem 106
620 IFCX=0ORCY=0ORCZ=0THEN700      :rem 74
630 IFCX=3THENVX=0:NX=X            :rem 242
640 IFCY=3THENVY=0:NY=Y            :rem 247
650 IFCZ=3THENVX=0:VY=0:NX=X:NY=Y  :rem 195

660 IFCX=6ORCY=6THEN780            :rem 183
670 POKEH+X+40*Y,113:POKEH+NX+40*NY,2+16*4 :rem 154
680 X=NX:Y=NY                      :rem 80
690 GOTO460                          :rem 114
700 POKEH+X+40*Y,113                :rem 100
710 IFAX<>0THENPOKEHX+AX,113        :rem 237
720 IFAY<>0THENPOKEHY+AY*40,113     :rem 127
730 FORT=700TO300STEP-20            :rem 170
740 SOUND3,T,2                      :rem 53
750 NEXT                             :rem 219
760 AX=0:AY=0:VX=0:VY=0:X=1:Y=1:NX=1:NY=1 :rem 45

770 GOTO440                          :rem 111
780 T=VAL(MID$(TI$,3,2))            :rem 49
790 PRINT"{HOME}HAI VINTO!! IN "T" MIN"; :rem 118

800 FORT=1TO6:READS,J:SOUND1,S,J:rem 34
810 FORM=1TO280:NEXT:NEXT           :rem 106
820 PRINT"{GIU'}GIOCHI ANCORA (S/N)?" :rem 67

830 GETKEYAS                         :rem 201
840 IFA$<>"S"THENPRINT"{CLR}CIAO!":END :rem 130

850 RUN                              :rem 146
860 DATA00000000000000000000000010001010000000 :rem 172
1000 :rem 172
870 DATA1111101111111111111110101010111111 :rem 199
1010 :rem 199
880 DATA000001000001000000000101010000010 :rem 177
0010 :rem 177
890 DATA01110101110101111111101011101110 :rem 198
1110 :rem 198

900 DATA000100000100000100010001000100000010 :rem 170
1000 :rem 170
910 DATA11011101010111110101010101111110 :rem 191
1111 :rem 191
920 DATA010001010101000101010101000100000 :rem 176
0000 :rem 176
930 DATA01110101010101010101010111110111011 :rem 191
1111 :rem 191
940 DATA010001010111010001010100010000010 :rem 179
0000 :rem 179
950 DATA000111010101011111010101010101110 :rem 191
1111 :rem 191
960 DATA01000101010000001000001010101000 :rem 178
0000 :rem 178
970 DATA011101110111010101110111000101010 :rem 193
1011 :rem 193
980 DATA01000000100010100010101111101010 :rem 185
1000 :rem 185
990 DATA0101111110111010111100000001010 :rem 194
1110 :rem 194
1000 DATA01010000000100010100010111111101 :rem 219
11000 :rem 219
1010 DATA0001111111101110101010100000101 :rem 225
01011 :rem 225
1020 DATA01010000000100000101010101010001 :rem 215
00000 :rem 215
1030 DATA01010111110101111101010101011101 :rem 230
01110 :rem 230
1040 DATA00010100000001000101010101010001 :rem 218
01000 :rem 218
1050 DATA0101011111111010111011101010111 :rem 235
01111 :rem 235
1060 DATA01010000000000010000000001000100 :rem 214
00000 :rem 214
1070 DATA59,65,73,151,157,215,231,281,303 :rem 202
,315,331,393,447,451,477 :rem 202
1080 DATA541,543,593,615,673,709,723,767, :rem 18
779,783,796,841,871 :rem 18
1090 DATA685,20,685,20,704,20,739,35,685, :rem 227
20,770,60 :rem 227

```



**GRUPPO
EDITORIALE
JACKSON**

LA BIBLIOTECA CHE FA TESTO.



La tana del dragone

di **C. Book**
trad. e adatt.
di **F. Sarcina**

Pochi uomini sono riusciti a sopravvivere ai pericoli del Castello di Dunsmore, dimora di molteplici dragoni. Un gioco di strategia per C16 e Plus/4.

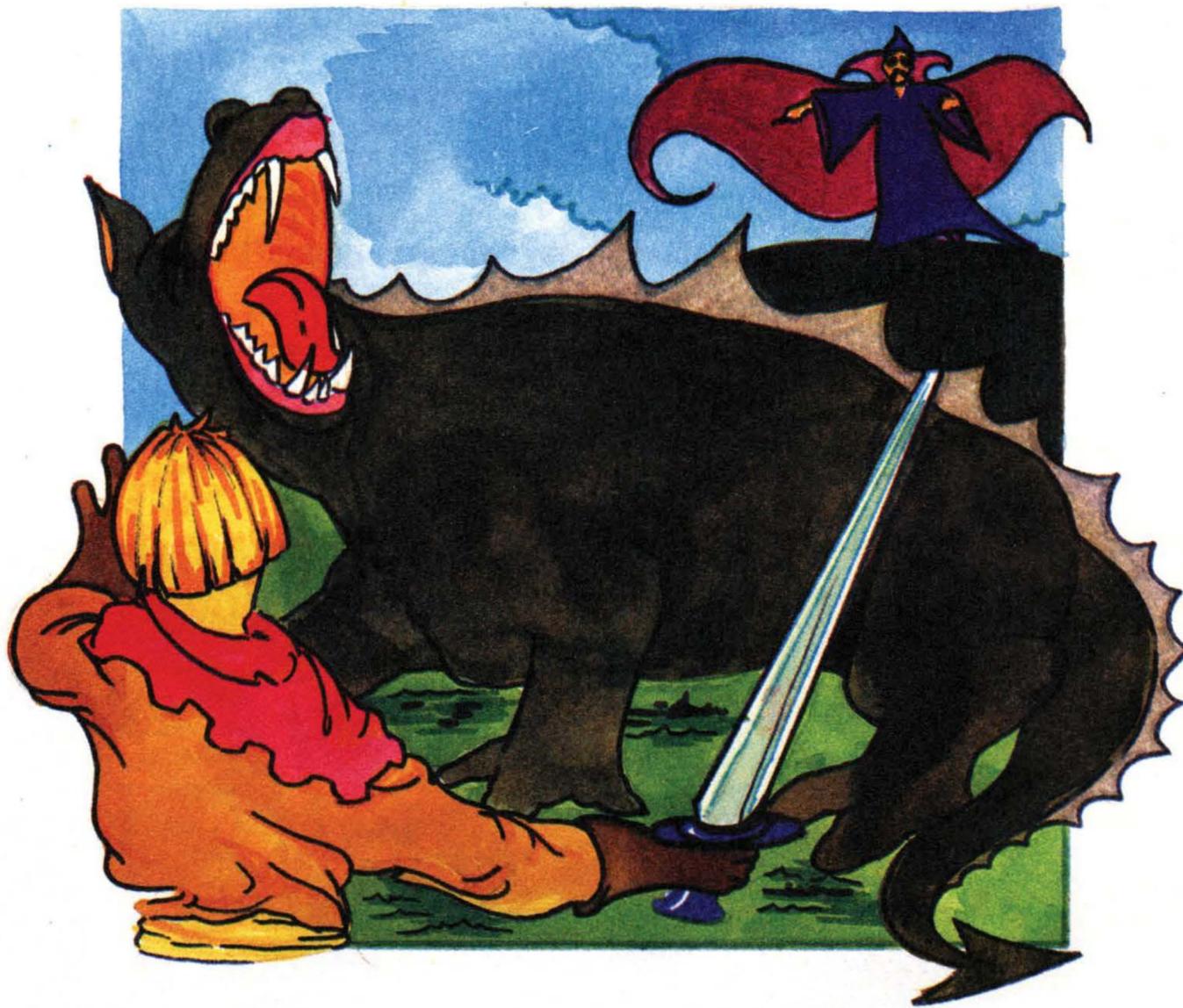
Trovare la strada per uscire dai tortuosi sotterranei del Castello di Dunsmore è difficile. Hai poco cibo e poca acqua a disposizione per sopravvivere, e se non sei più che lesto di gambe, rischi un'orribile fine nelle fauci di un dragone.

Come si gioca

LA TANA DEL DRAGONE è un gioco scritto interamente in BASIC. Dopo aver digitato il programma salvane una copia. Dopo il RUN appare un messaggio di attesa: il computer sta ora disegnando il labirinto dei sotterranei del castello. Ti troverai quindi in una stanza di questi sotterranei. Lo schermo presenta una visione tridimensionale prospettica della stanza e un indicatore direzionale verso il Nord. A questo punto hai diversi comandi da scegliere indicati nella parte bassa dello schermo: per selezionare uno di essi premi la prima lettera della parola (per sinistra, ad esempio). In ogni stanza sono sempre disponibili almeno quattro delle seguenti opzioni:

CARTA: permette di vedere la tua attuale posizione nei sotterranei del castello e i posti in cui sei già stato. Indica con una M anche i luoghi dove hai lasciato del pane.

PANE: deposita un pezzo di pane nella stanza. Appare un messaggio che indica i pezzi di pane utilizzati (all'inizio se ne hanno a disposizione 20). Serve per ricordarsi posti particolari che si pensa possano servire in futuro. Tuttavia, se utilizzi tutti i 20 pezzetti di pane, muori di fame.



IND. (indietro): per spostarsi nella camera immediatamente indietro a quella in cui si è al momento.

SIN. (sinistra): per girare a sinistra (ovviamente solo se verso sinistra vi è un altro corridoio).

AV. (avanti): per proseguire dritto.

DESTRA: per girare a destra.

BOTOLA: solleva il coperchio della botola presente nella stanza. Sotto potre-

sti trovare una spada magica, oppure dell'oro, o dell'acqua. O anche un orribile e affamato dragone.

Consigli di gioco

Lo scopo di LA TANA DEL DRAGONE è riuscire a fuggire dal castello, magari dopo aver racimolato un buon bottino in oro. Bisogna quindi trovare un'uscita. Controlla regolarmente la posizione con l'opzione Carta: l'uscita è

lungo le mura esterne. Sembra facile, ma trovarla richiede senz'altro un certo tempo, e con il tempo finisce anche l'acqua a tua disposizione; allora dovrai comunque aprire delle botole, dove potrai trovare dell'acqua.

Nelle botole puoi anche trovare pepite d'oro, o una spada che ti serve per uccidere un drago, ma che funziona per una volta sola. Se dovessi incontrare un drago non avendo la spada a disposizione, dovrai fuggire velocemente, per

evitare di fare una cruenta e orribile fine nelle fauci del drago. Se incontri un'uscita, ma ritieni di non aver accumulato abbastanza oro puoi anche decidere di rimanere ancora nel labirinto.

La tana del drago

```

100 Y=826:COLOR4,1:COLOR0,1:PRINT" {CLR}
    { 5 GIU' } [<5>] { 10 SPAZI } LA TANA DEL
    DRAGONE" :rem 104
110 PRINT" { 10 GIU' } ATTENDERE, STO COSTR
    UENDO IL LABIRINTO { RED }" :rem 38
120 BO$=" { 39 SPAZI }":X$="{HOME}
    { 23 GIU' }" :rem 154
130 F=831:G=832:H=829:C=3:DIMM$(10,11):FO
    RI=1TO9:FORJ=1TO10 :rem 179
140 M$(I,J)=STR$(INT((7*RND(1))+1))+STR$(
    INT((8*RND(1))+1)):NEXTJ,I :rem 243
150 S=I+10:I=INT((8*RND(1))+1):J=INT((9*R
    ND(1))+1):GOTO520 :rem 233
160 PRINTX$"{DES} [<4>]";B$:B$="":IFY$="P"
    THENPRINTTAB(18)" {SU}"SL :rem 50
170 PRINT" [<8>] { 2 SU } {RVS}C{OFF}ARTA
    {RVS}P{OFF}ANE {RVS}I{OFF}ND.";SS$"
    {RED}":IFHS=1THEN1320 :rem 214
180 IFWT>100ORS<1THEN1180 :rem 252
190 GETY$:IFY$=""THEN190 :rem 135
200 IFDE=1ANDY$<>"I"THEN190 :rem 56
210 GOSUB1580:DE=0:PRINT" {SU}"SB$,SB$"
    {SU}":GOSUB1560:IFY$="C"THEN1420
    :rem 166
220 IFY$="P"THENGOSUB1530:GOSUB930:SL=SL+
    1:S=S-1:B$="PEZZI DI PANE":GOTO160
    :rem 57
230 IFTC=1THENTC=0 :rem 92
240 WT=WT+1:IFY$="S"ANDLD=1ORY$="D"ANDRD=
    1ORY$="A"ANDEFD=1ORY$="I"THEN280
    :rem 134
250 IFY$="B"THENPRINT" {HOME} { 13 GIU' }"SP
    C(20)" {SU}" :rem 196
260 IFY$="B"ANDTD=1THEN960 :rem 14
270 GOTO160 :rem 105
280 SAF=I:FAS=J:TD=0:LD=0:RD=0:FD=0:IFY$=
    "I"THENTC=1 :rem 189
290 IFTB=1THEN470 :rem 248
300 IFLT=1THEN370 :rem 249
310 IFRT=1THEN420 :rem 252
320 TB=0:LT=0:RT=0:IFY$="S"THENJ=J-1:LT=1
    :C=1 :rem 215
330 IFY$="D"THENJ=J+1:RT=1:C=2 :rem 253
340 IFY$="A"THENI=I-1:C=3 :rem 174
350 IFY$="I"THENI=I+1:TB=1:C=4 :rem 244
360 GOTO510 :rem 104
370 TB=0:LT=0:RT=0:IFY$="S"THENI=I+1:TB=1
    :C=4 :rem 209
380 IFY$="D"THENI=I-1:C=3 :rem 181
390 IFY$="A"THENJ=J-1:LT=1:C=1 :rem 251
400 IFY$="I"THENJ=J+1:RT=1:C=2 :rem 0
410 GOTO510 :rem 100
420 TB=0:LT=0:RT=0:IFY$="S"THENI=I-1:C=3
    :rem 144
430 IFY$="D"THENI=I+1:TB=1:C=4 :rem 238
440 IFY$="A"THENJ=J+1:RT=1:C=2 :rem 252
450 IFY$="I"THENJ=J-1:LT=1:C=1 :rem 0
460 GOTO510 :rem 105
470 TB=0:LT=0:RT=0:IFY$="S"THENJ=J+1:RT=1
    :C=2 :rem 226

```

```

480 IFY$="D"THENJ=J-1:LT=1:C=1 :rem 254
490 IFY$="A"THENI=I+1:TB=1:C=4 :rem 241
500 IFY$="I"THENI=I-1:C=3 :rem 180
510 SS$="":GOTO790 :rem 244
520 PRINT" {CLR}":FORIJ=1TO20:PRINT"
    [< 15 +>]"SPC(10)" [< 14 +>]":NEXTIJ
    :rem 227
530 PRINT" {HOME}":FORII=1TO4:PRINTTAB(15)
    "{RVS} [<R>] [<E>] [<R>] [<E>] [<R>] [<E>]
    [<R>] [<E>] [<R>] [<E>] {GIU'} { 10 SIN }
    [<E>] [<R>] [<E>] [<R>] [<E>] [<R>] [<E>]
    [<R>] [<E>] [<R>]":NEXTII :rem 59
540 PZ=12:ZP=15:FORIJ=1TO12:ZP=ZP-1:FORJI
    =1TOPZ :rem 42
550 PRINTTAB(ZP)" ";:NEXTJI:PZ=PZ+2:PRINT
    :NEXTIJ:PRINT" {HOME} { 8 GIU' }"
    :rem 37
560 JJ=8:FORII=14TO3STEP-1:JJ=JJ+2
    :rem 101
570 PRINTTAB(II)"_ "SPC(JJ)" [<*>]":NEXTII
    :rem 201
580 PRINTLEFT$(X$,19)TAB(16)" { 4 SPAZI }":
    PRINTTAB(17)" { 4 SPAZI }":PRINTLEFT$(X
    $,19); :rem 254
590 IFC=3THENPRINTTAB(17)" [<8>]N{RED}":PR
    INTTAB(17)"↑" :rem 110
600 IFC=4THENPRINTTAB(17)"V":PRINTTAB(17)
    " [<8>]N{RED}" :rem 95
610 IFC=1THENPRINTTAB(16)" [<8>]C{RED}N"
    :rem 108
620 IFC=2THENPRINTTAB(16)"N [<8>]C{RED}"
    :rem 108
630 IFI>9ORJ>10ORI<1ORJ<1THENSF=I:FAS=J:
    GOTO760 :rem 91
640 MM=VAL(LEFT$(M$(I,J),2)):ONMMGOTO650,
    660,670,680,690,700,710,760 :rem 223
650 GOSUB850 :rem 184
660 GOSUB770 :rem 186
670 GOSUB880:GOTO720 :rem 201
680 GOSUB850 :rem 187
690 GOSUB770:GOTO720 :rem 201
700 GOSUB880 :rem 183
710 GOSUB850 :rem 181
720 IFRIGHT$(M$(I,J),2)="22"THENGOSUB930
    :rem 42
730 IFMM>4THENGOSUB940 :rem 129
740 IFVAL(MID$(M$(I,J),4,1))<6THENGOSUB12
    60 :rem 213
750 GOTO160 :rem 108
760 GOSUB1580:PRINTX$SSPC(6);"{SU}{RVS}STR
    ADA CIECA{OFF} TORNA {RVS}I{OFF}NDIET
    RO!":DE=1:GOTO180 :rem 71
770 PRINTLEFT$(X$,10):FORGI=1TO4:PRINTTAB
    (18)" { 2 SU } { 4 SPAZI }":NEXT:rem 1
780 PRINTTAB(18)" { 2 SU }_ { 2 SPAZI } [<*>]"
    :FD=1:PQ=19:SS$=SS$+" {RVS}A{OFF}V.":
    GOTO920 :rem 80
790 PRINT" {HOME}":FORII=1TO4:PRINTTAB(15)
    "{RVS} [<R>] [<E>] [<R>] [<E>] [<R>] [<E>]
    [<R>] [<E>] [<R>] [<E>] {GIU'} { 10 SIN }
    [<E>] [<R>] [<E>] [<R>] [<E>] [<R>] [<E>]
    [<R>] [<E>] [<R>]":NEXTII :rem 67

```

```

800 PRINT"{HOME}{ 8 GIU' }":FORII=1TO8:PRI
NTTAB(15)"{ 10 SPAZI }":NEXTII
:rem 164
810 PRINT"{HOME}{ 4 GIU' }":FORGI=1TO9:PRI
NTTAB(6)"[< 4 +>]":NEXT:LD=0:PRINTTAB
(6)"[< 3 +>]E" :rem 113
820 PRINTTAB(6)"[< 2 +>]E":PRINTTAB(6)"
[<+>]E":PRINTTAB(6)"E" :rem 136
830 PRINT"{HOME}{ 4 GIU' }":FORGI=1TO9:PRI
NTTAB(30)"[< 4 +>]":NEXT:RD=0:PRINTTA
B(30)"[<*>][< 3 +>]" :rem 9
840 PRINTTAB(31)"[<*>][< 2 +>]":PRINTTAB(
32)"[<*>][<+>]":PRINTTAB(33)"[<*>":G
OTO580 :rem 201
850 PRINTLEFT$(X$,19):FORGI=1TO3:PRINTTAB
(6)"{ 2 SU}{ 4 SPAZI }":NEXT :rem 213
860 FORGI=1TO8:PRINTTAB(6)"{ 2 SU}
[< 4 T>]":NEXT:LD=1 :rem 38
870 SS$=SS$+" {RVS}S{OFF}IN.":QP=6:PQ=7:G
OTO910 :rem 69
880 PRINTLEFT$(X$,19):FORGI=1TO3:PRINTTAB
(30)"{ 2 SU}{ 4 SPAZI }":NEXT:RD=1
:rem 67
890 FORGI=1TO8:PRINTTAB(30)"{ 2 SU}
[< 4 T>]":NEXT:LD=1 :rem 86
900 SS$=SS$+" {RVS}D{OFF}ESTRA":QP=30:PQ=
31 :rem 55
910 PRINTTAB(QP)"{ 2 SU}E[< 2 T>][<*>]"
:rem 36
920 PRINTTAB(PQ)"{ 2 SU}E[<*>]":RETURN
:rem 249
930 PRINTLEFT$(X$,16)TAB(20)"[<8>]U*I
{GIU'}{ 3 SIN}[<Z>]*[<X>]{RED}":RETUR
N :rem 231
940 PRINTLEFT$(X$,13)TAB(17)"N[< 5 T>]M":
PRINTTAB(16)"N{ 3 SPAZI}W{ 3 SPAZI}M
" :rem 128
950 PRINTTAB(16)"[< 9 U>]":TD=1:SS$=SS$+"
{RVS}B{OFF}OTOLA":RETURN :rem 175
960 PRINT"{HOME}{ 9 GIU' }"TAB(16)"
[< 9 O>]":PRINTTAB(16)"M{ 7 SPAZI}N"
:rem 165
970 PRINTTAB(17)"M{ 5 SPAZI}N" :rem 156
980 SS=LEN(SS$):SS$=LEFT$(SS$,SS-6)
:rem 94
990 FE=FE+1 :rem 81
1000 ONINT((9*RND(1))+1)GOTO1150,1010,101
0,1150,1030,1010,1140,1010,1160,1030
:rem 129
1010 B$="{RVS}SENTI IL RUMORE DELLE SQUAM
E!":HS=1:PRINTLEFT$(X$,14)TAB(18)"*
*" :rem 224
1020 GOTO160 :rem 147
1030 IFFE<20THEN1010 :rem 109
1040 Y$="":PRINTX$"{SU}{ 3 SPAZI}[<8>]TRO
VI UN'USCITA! {RVS}S{OFF}CAPPI O
{RVS}R{OFF}IMANI?":FE=5 :rem 75
1050 GETY$:IFY$=""THEN1050 :rem 223
1060 IFY$="R"THENPRINT"{SU}"SB$:GOTO160
:rem 250
1070 IFY$<>"S"THEN1040 :rem 217
1080 PRINT"{CLR}{ 7 GIU'}[<8>]{ 4 SPAZI}
HAI TROVATO UN'USCITA, E FUGGI!":IFG
L=0THEN1200 :rem 146
1090 PRINT"{GIU'}{ 10 SPAZI}CON";GL;"PEPI
TE D'ORO!":GOTO1200 :rem 82
1100 IFSW=0THENGOSUB1400 :rem 220
1110 IFSW=0THENPRINT"{CLR}{ 6 GIU'}[<8>]
{ 4 SPAZI}SEI STATO MANGIATO DA UN D
RAGONE":GOTO1200 :rem 23

```

LOAD

LA TANA DEL DRAGONE

```

1120 B$="{RVS}HAI TROVATO UN DRAGONE E US
ATO LA SPADA":SW=0 :rem 121
1130 PRINTLEFT$(X$,14)TAB(18)"++":GOTO16
0 :rem 52
1140 B$="{RVS}C'E' DELL'ACQUA, GUADAGNI T
EMPO":WT=0:GOTO160 :rem 149
1150 B$="{RVS}NON C'E' NIENTE":GOTO160
:rem 78
1160 IFSW=1THENB$="{RVS}HAI TROVATO DELL'
ORO!":GL=GL+1:GOTO160 :rem 244
1170 B$="{RVS}HAI TROVATO UNA SPADA MAGIC
A!":SW=1:GOTO160 :rem 47
1180 IFS<1THENPRINT"{CLR}{ 4 GIU'}[<8>]
{ 5 SPAZI}SPIACENTE - SEI MORTO DI F
AME!":GOTO1200 :rem 195
1190 PRINT"{CLR}{ 4 GIU'}[<8>]{ 5 SPAZI}
SPIACENTE - SEI MORTO DI SETE!":GOTO
1200 :rem 94
1200 FORIJ=1TO100:NEXT:POKE59467,0
:rem 44
1210 PRINT"{ 4 GIU'}{ 10 SPAZI}VUOI RIPRO
VARE (S/N)":AG$="" :rem 94
1220 GETAG$:IFAG$=""THEN1220 :rem 59
1230 IFLEFT$(AG$,1)="S"THENRUN :rem 246
1240 IFLEFT$(AG$,1)<>"N"THEN1220:rem 255
1250 END :rem 159
1260 ONVAL(MID$(M$(I,J),4,1))GOSUB1270,12
80,1290,1300,1310:RETURN :rem 91
1270 B$="{RVS}L'ODORE DI FUMO E' FORTE":R
ETURN :rem 121
1280 B$="{RVS}FULIGGINE SULLE PARETI":RET
URN :rem 194
1290 B$="{RVS}SI SENTE DELL'ACQUA CHE GOC
CIOLA!":RETURN :rem 223
1300 B$="{RVS}SANGUE SUL PAVIMENTO":RETUR
N :rem 65
1310 B$="{RVS}RIMBOMBA UNO SBATTERE D'ALI
":RETURN :rem 174
1320 VB=55:Y$="":HS=0:TD=0 :rem 190
1330 GETY$:VB=VB-1:IFVB<45THEN1380
:rem 42
1340 FORIJ=1TO7:NEXT :rem 3
1350 IFY$="S"ANDLD=1ORY$="D"ANDRD=1ORY$="
A"ANDFD=1ORY$="I"THEN1370 :rem 193
1360 GOTO1330 :rem 202
1370 GOTO210 :rem 151
1380 IFSW>0THEN1120 :rem 102
1390 IFSW<=0THEN1100 :rem 160
1400 PRINTLEFT$(X$,12)TAB(19)"X*X":PRINTT
AB(18)"(* *)":PRINTTAB(18)"JQK"
:rem 134
1410 PRINTTAB(17)"<<<<>>>>":FORIJ=1TO3000
:NEXT:RETURN :rem 42

```

```

1420 SS$="":PRINT {CLR}{PUR}"TAB(14)"
      [< 10 @>]" :rem 133
1430 FORIJ=1TO9:PRINTTAB(13)"{OFF}{<M>}";
      :FORJI=1TO10 :rem 189
1440 MM$=RIGHT$(M$(IJ,JI),2):IFIJ=IANDJI=
      JTHENPRINT "{OFF}↑";:GOTO1480
      :rem 99
1450 IFMM$="-2"THENPRINT "{OFF}L";:GOTO148
      0 :rem 208
1460 IFMM$="22"THENPRINT "{RVS}M";:GOTO148
      0 :rem 215
1470 PRINT "{RVS} "; :rem 234
1480 NEXTJI:PRINT "{OFF}{<G>}":NEXTIJ:PRIN
      TTAB(14)"[< 10 T>]{RED}" :rem 255
1490 PRINT "{ 3 GIU' } { 7 SPAZI } {WHT}PREMI
      UN TASTO PER TORNARE" :rem 132
1495 PRINTTAB(12)"{GIU'}NEI SOTTERRANEI
      {RED}" :rem 136
1500 GETY$:IFY$=""THEN1500 :rem 223
1510 IFRIGHT$(M$(I,J),2)="-2"ORRIGHT$(M$(
      I,J),2)="22"THEN520 :rem 38
1520 M$(I,J)=M$(I,J)+"-2":GOTO520
      :rem 223
1530 IFRIGHT$(M$(I,J),2)="-2"THENM$(I,J)=
      M$(I,J)+"2":RETURN :rem 51
1540 IFRIGHT$(M$(I,J),2)="22"THENRETURN
      :rem 31

```

```

1550 M$(I,J)=M$(I,J)+"22":RETURN:rem 247
1560 IFRIGHT$(M$(I,J),2)="-2"ORRIGHT$(M$(
      I,J),2)="22"THENRETURN :rem 116
1570 M$(I,J)=M$(I,J)+"-2":RETURN:rem 244
1580 PRINTLEFT$(X$,23);BO$:PRINTX$BO$:RET
      URN :rem 10

```



**Dalla grande
edicola
Jackson**

**Tutto sul personal
computer**

COMPUSCUOLA

La rivista di informatica nella didattica per la
scuola italiana.

9 numeri all'anno: L. 3.000 a numero

Abbonamento: solo L. 20.000

**medical
computer**

Una novità assoluta per l'editoria
tecnico-scientifica, studiata appositamente
per i medici, la Rivista di personal computer,
affari personali e tempo libero.

11 numeri all'anno: L. 5.000 a numero

Abbonamento: solo L. 45.000

Bit

La prima rivista europea di personal
computer, software e accessori. Con test,
novità, analisi del mercato ...

11 numeri all'anno: L. 5.000 a numero

Abbonamento: solo L. 43.000

PC

L'unica rivista italiana dedicata ai sistemi
MS-DOS, Personal computer IBM e
compatibili.

11 numeri all'anno: L. 5.000 a numero

Abbonamento: solo L. 44.000

**Quando l'informazione
fa testo**

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

Desidero ricevere GRATIS un numero
della Rivista _____

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

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

Nome _____

Cognome _____

via _____

CAP _____ Città _____

input.output



**Piccoli
annunci**

Vendo, cambio programmi di tutti i generi per C 64. Ne possiedo circa 200 e sono tutti novità assolute. Invio la lista a chi mi invia la sua. Massima serietà; rispondo a tutti. Francesco Palmieri - Via Porta Roma, 4 - 81011 Alife (CE) - Tel. 0823/918954

Vendo C 64 in ottime condizioni e stato d'uso con cassetta giochi a L. 250.000 trattabili, perché intendo acquistare il C 128. Mario Gianmarco - Via Calamandrei, 5 - 48022 Lugo (RA) - Tel. 0545/22965

Vendo, cambio programmi per il C 128 e C 64, sia su disco che su cassetta. Dispongo delle ultime novità. Per maggiori informazioni scrivere a: Massimo Tabasso - Piazza Molineris, 1 - 12038 Savigliano (CN)

Vendo per C 64 programma per progettazione e dimensionamento di impianti idrici antincendio corredato da istruzioni e tabelle. Scrivere o telefonare a: Felice Venuti - Via S.Marta, 316 Pal. H - 98100 Messina - Tel. 090/713906

Vendo C 16 più registratore, 3 cassette e joystick a sole L. 230.000. Sì, avete letto bene, solo L. 230.000 Andrea Pucci - Via Lividonia, 63 - 58019 Porto S.Stefano (GR) - Tel. 0564/813259

Vendo C 16 più registratore, joystick, adattatore, manuale, 10 riviste, 20 giochi e due utility a sole L. 280.000. Benito Pontiroli - Via C. Terranova, 57 - 41100 Modena - Tel. 059/391639

Per C 64 vendo cartuccia Fast load a L. 35.000 (è utilissima). Inoltre vendo espansione memoria 8-16 Kbyte per VIC 20. Cambio o vendo tantissimo software per C 64. Telefonare dalle ore 18.00 alle ore 21.00. Inviare liste e richieste a: Gianni Cottogni - Via Strambino, 23 - 10010 Carrone (TO) - Tel. 0125/712311

Cambio i miei giochi su disco per C 64 (oltre 1.000) con altri che non ho. A chi lo richiede invio lista gratuita per scambi. Franco Rigamonti - Casella Postale, 26 - 22061 Barzago (CO) - Tel. 039/957091

Vendo Plus/4 nuovo, 3 mesi di vita, a L. 500.000 con programmi vari. Oppure cambio con C 64 nuovo. Luigi Bricchi - Via De Amicis, 4 - 21033 Cittiglio (VA) - Tel. 0332/602130

Vendo C 64 nuovissimo, mai tirato fuori dagli imballi, a L. 300.000. Inoltre vendo C 64 usato a L. 250.000. In regalo vasto assortimento dei migliori video game in commercio. Diego Zanin - Via Di Vittorio, 25 - 10093 Collegno (TO) - Tel. 011/4150446

VIC 20 software club ha tutti i programmi per dimostrarvi che potete usare il VIC 20 al pari degli altri più reclamizzati. VIC 20 software club c/o Cinzia Ceccarini - Via Di Vittorio, 10 - 58022 Follonica (GR)

Vendo C 128 più manuale, disco originale CP/M plus 30 a L. 650.000. Regalo natalizio indesiderato. Garanzia un anno. Rinaldo Viglione - Via Ginestrato, 3L/3scs - 16144 Genova - Tel. 010/821284

Vendo programmi per C 64 su cassetta (più di 300) da L. 500 a L. 1.500. Richiedere la lista a: Marco Romani - Via G. Valli, 95 - 00149 Roma - Tel. 06/5265121

Cambio e vendo programmi per C 64: Haker, Brunòs boxing, Lode runner, Who dares wins, Nexus, Strip Poker 2, Impossible mission, altri games e parecchie utility sia su disco che su nastro (la maggior parte su disco). Inviatemi la vostra lista, vi invierò la mia. Luca Rovatti - Via B. Migliarina, 97 - 41012 Carpi (MO) - Tel. 059/661922

Vendo Commodore 64 a L. 360.000 (un anno di vita) con imballo e istruzioni originali; in regalo 25 programmi a scelta in LM (su disco o cassetta). Massima serietà. Oscar Sillani - Via Attilio Greco, 5 - 84090 Montecorvino Pugliano (SA) - Tel. 089/801305

Personalizzo, correggo programmi per C 64 in BASIC o LM. Compro, vendo o cambio software. Annuncio sempre valido. Inviare lista o prendere accordi con: Giovanni Catalano - Via Interna Marina, 19 - 88074 Crotona (CZ) - Tel. 0962/61413

Dispongo di circa 600 programmi tra giochi e utility su cassetta per C 64. Telefonare verso sera o festivi per ricevere la lista. Massimo Di Carlo - Via Maffi, 112/G - 20099 Sesto S. Giovanni (MI) - Tel. 02/2425618

Scambio, acquisto, vendo programmi per C 64. Scrivere o telefonare a: Gianni Negrello - Via Val Frenzela, 38 - 36020 Valstagna (VI) - Tel. 0424/90157 (dalle 8.30 alle 14.30) - serietà e simpatia.

Vendo o cambio programmi per C 64 su disco e cassetta, cerco utility per drive 1541. Arrivi settimanali, ultime novità. Marco Mercanti - Via Pio Majaloni, 76 - 00159 Roma - Tel. 02/4373763

Per C 64 cambio software di ogni genere. Inviatemi le vostre liste o telefonatemi. Non intendo acquistare o vendere, solo scambiare. Maurizio Garutti - Via Paolini, 12 - 10138 Torino - Tel. 011/441278

Compro espansione per C 64. Telefonare a: Andrea Sonaglia - Via P. Neruda, 60 - 06073 Perugia - Tel. 075/6978347

Vendo videogame (Raid over Moscow, Tutto il calcio minuto per minuto, Golf) per C 64. Richiedete la lista, ve la spedirò gratis. Maximiliano Magri-Tilli - Via Pace, 270 - 62100 Macerata

Compro manuali o fotocopie dei seguenti programmi: Easy script; Easycalc, Music maker, Magic desk I, Koalainter, Gesmag, Simons' BASIC, Super64-Mon, Magic key, Disk mimic 64, Superbase 64, Tot professional, Preasy script, Mailing list. Pierangelo Bellini - Via F. Turati, 15 - 01016 Tarquinia (VT) - Tel. 0766/855736

Cambio programmi solo su cassetta per C 16. Telefonare solo dopo le ore 20.00. Massimiliano Favero - Via Garrian, 57 - 20146 Milano - Tel. 02/4221411

Per C 64 vendo al miglior offerente 150 dischetti pieni di non meno di 1.000 game (i migliori) oppure vendo singolarmente a prezzi stracciati. Maurizio Anfosso - Via Simonetti - 14018 Roatto d'Asti (AT) - Tel. 0141/938118

Vendo ottimi videogiochi per C 64. Dispongo dei migliori: Calcio con replay, Ghostbusters, Bruce Lee, Boxe, Pit-stop II, ecc. Vendo cassetta da 10 - 20 - 50 - 100 giochi a partire da L. 7.000. Massima serietà. Danilo Autullo - Via Cavalieri di Vittorio Veneto, 17 - 00049 Velletri (RM) - Tel. 06/9634980

Compro programmi solo su disco per C 64 (giochi, utility), anche ad alto prezzo. Inviare lista o telefonare. Giulio Bodrato - Via A. Barelli, 5 - 20148 Milano - Tel. 02/401675

Cerco stampante Commodore MPS802. Inviare offerte a: Alessandro Girardi - Via E. Barbaro, 5 - 31100 Treviso - Tel. 0422/260273

Vendo o cambio per C 64 cassetta contenente 20 game (Zaxxon, Pole, position, Karate) a sole L. 8.000 e cassette con 18 utility e 12 game (Koala painter, Simons' BASIC, Copy 190) a sole L. 15.000. Vendo programmi a L. 800. Marco Baldessari - Via Baldessari, 14 - 38041 Albiano (TN) - Tel. 0461/689628

Cambio user manual C 16 (nuovo) con user manual VIC 20, anche usato, purché completo. Spese postali alla pari. Claudio Fabietti - Località Borda si Sersale - 88050 Cropani Marina (CZ) - Tel. 0961/961268

Per C 64 vendo sprotettore universale ed efficace nel 95% dei casi a L. 25.000, su nastro o disco. Massima serietà. Giuseppe Rinanti - Via Oreto, 332 - 90124 Palermo - Tel. 091/444664

Vendo kit programmi BASIC da me creati per dotare ogni cassetta per C 64 di indice con ricerca e caricamento programmi BASIC e LM. Kit, istruzioni dettagliate e listati a L. 20.000 più spese di spedizione. Scrivere oppure telefonare la sera. Riccardo Pratesi - Via Fiume, 163 - 19100 La Spezia - Tel. 0187/22387

Vendo Commodore Plus/4, ancora imballato, a L. 400.000 (senza registratore). Telefonare ore pasti. Dino Olivieri - Via Tofane, 2 - 10141 Torino - Tel. 011/380341

Eccezionale: più di 1.000 programmi gratis iscrivendovi al Commodore Computer Club. Alcuni titoli: Dragons lair, Winter games, Silent service. Per averli scrivete a: Commodore Computer Club - Via Calatafimi, 15 - 91026 Mazara del Vallo (TP)

Vendo Commodore VIC 20 più espansione da 3 Kbyte, quattro cassette giochi, joystick Commodore e tre libri giochi a L. 95.000. Rinaldo Cuccurullo - Via A. De Gasperi, 131 - 80059 Torre del Greco (NA) - Tel. 081/8810149

Cambio, vendo circa 3.000 programmi per C 64. Vendo anche cartuccia Isepick e Turbo drive. Per VIC 20 cedo espansione memoria da 8-16 Kbyte. Telefonare dalle 18.00 alle 21.00. Gianni Cottogni - Via Strambino, 23 - 10010 Carrone (TO) - Tel. 0125/712311

Vendo programmi per C 64 a L. 2.500 cadauno. Telefonare a: Augusto Tonucci - Viale Romagna, 36 - 61032 Fano (PS) - Tel. 0721/879424

Vendo VIC 20 più espansione 16 Kbyte e cinque cartucce tra cui "Scacchi". Tutto, in buone condizioni, a L. 300.000: un vero affare. In regalo 25 giochi. Guido Bonino - Via delle Rose, 4 - 10090 Villarbasse (TO) - Tel. 011/952333

Vendo software per C 64; tutte le ultime novità a vostra disposizione, tra le quali: Zorro, Rambo, Robin Hood, The goonies, Wargames (Coleco), Kung fu master, Karatè, Champ, The eidolon, Koronis rift, Never ending story e moltissimi altri Maurizio Porzionato - Via Avogadro, 22 - 13010 Quaregna (VC) - Tel. 015/94751

Causa passaggio a sistema superiore, vendo C 64 più 200 programmi di vario genere a L. 550.000 non trattabili. Preferibilmente zona Iglesias. Gianluca Diana - Via Agruxau, 8 - 09016 Iglesias (CA)

Vendo per C 16 alcuni giochi tra i quali Calcio, Olimpiadi, Frogger, Space invaders, Berks, ecc. Mario Tarable - Via Vittorio Emanuele, 6 - 12051 Alba (CN) - Tel. 0173/42214

Cambio o vendo software per VIC 20 inespanso ed ed panso in cassetta e in cartridge. Tratto con la zona di Torino. Marco Guglielmina - Corso Orbassano, 255/C - 10137 Torino - Tel. 011/395702

Vendo o cambio stupendi giochi come Raider over Moscow, Il ritorno delle spie, Rollerball, Il paesaggio incantato e tanti altri, ancora più belli, a L. 500 cadauno. L'acquirente riceverà inoltre Turbo tape II (dura 7 giri). Cosimo Petracca - Via Livorno, 148 - 85100 Potenza - Tel. 0971/43035

Recensioni Libri

I più venduti

Titolo	Autore	Cod.
Lavoriamo con il Commodore 16 (con cassetta)	A. Borra	CC244
Guida al Commodore Plus 4	M. Cristuib Grizzi R. Bonelli L. Pazzucconi F. Racchi	CC256
Commodore 16 per te: Basic 3.5 (con cassetta)	R. Bonelli	413B
Commodore 16 sempre di più (con cassetta)	R. Bonelli	427B
Commodore 16: essenziali routine (con cassetta)	D. Lawrence	422B

Biblioteca Jackson per Commodore 16-Plus/4-Vic 20

Cod.	Titolo	Autore	Prezzo
338D	Alla scoperta del VIC 20	R. Bonelli	22.000
413B	Commodore 16 per te: Basic 3.5 (con cassetta)	R. Bonelli	35.000
422B	Commodore 16: essenziali routine (con cassetta)	D. Lawrence	29.000
427B	Commodore 16 sempre di più (con cassetta)	R. Bonelli	35.000
553D	Giochi, trucchi e conti: 29 programmi per VIC 20	M. Benelfoul	24.000
557D	Giochi, giochi, giochi per il vostro VIC 20	M. Ramshaw	9.000
563D	Sinfonia per un computer VIC 20	T. Hartnell	10.000
565D	Sinfonia per un computer VIC 20 (con cassetta)	T. Hartnell	24.000
568D	Giochi, giochi, giochi per il vostro VIC 20 (con cassetta)	M. Ramshaw	24.000
CC244	Lavoriamo con il Commodore 16 (con cassetta)	A. Borra	20.000
CC256	Guida al Commodore Plus/4	M. Cristuib Grizzi R. Bonelli L. Pazzucconi F. Racchi	30.000

256 Rita Bonelli - Luciano Pazzucconi - Fabio Racchi
Guida al Commodore Plus/4
Pagine 480 - L. 30.000
cod. CC256

Un nuovo libro per i numerosi appassionati Commodore, rivolto in particolare modo a tutti gli utilizzatori del "giovane" Plus 4. Questo nuovo libro, per essere

sfruttato appieno, non richiede nessuna particolare conoscenza di programmazione. L'intento principale è quello di essere una guida per l'utilizzo del Commodore Plus/4 e per l'apprendimento della programmazione in linguaggio BASIC e in linguaggio ASSEMBLER.

Nel testo viene presentata l'implementazione del BASIC 3.5 disponibile su questa macchina; è illustrato il DOS residente nell'unità a floppy disk 1541 e

viene insegnato l'uso completo della stampante grafica MPS-803. Lo studio dell'ASSEMBLER è sviluppato appoggiandosi alle possibilità offerte dal comando MONITOR del BASIC, ma viene affrontato anche il problema del linguaggio macchina della CPU 7501 e della sua struttura hardware.

Le spiegazioni teoriche sono sempre accompagnate da esempi commentati, e per questo il libro può essere utile anche ai principianti.

Non resta altro da aggiungere se non che si tratta di un libro che può essere definito una guida completa per chi ha scelto il Plus/4.

244 Alessandro Borra - Mauro Cristuib Grizzi

Lavoriamo con il Commodore 16 (con cassetta)
Pagine 196 - L. 20.000
cod. CC244

Molto spesso gli home computer vengono acquistati come strumenti per il divertimento di ragazzi, piccoli e un po' cresciuti, senza poi sfruttarne appieno tutte le capacità. È un vero peccato possedere una macchina che offre diverse opportunità, come il Commodore 16, e non conoscerne le potenzialità.

Proprio per introdurre anche il neofita all'utilizzo del Commodore 16 con procedure professionali, il Gruppo Editoriale Jackson pubblica questo nuovo testo, dedicato a tutti i possessori (o utenti) di questo home computer.

Vengono quindi presentati tre programmi base per l'inserimento del computer in ambiente di lavoro o per un uso domestico più "intelligente".

Un word processor, programma di scrittura, interamente in ASSEMBLER, con il quale è possibile sostituire in modo totale la macchina da scrivere.

Un data base, che permette di gestire qualsiasi tipo di informazione, creare un'agenda e organizzare un archivio. Nel libro sono contemplate due versioni del programma, in modo da essere utilizzate indipendentemente dal tipo di supporto di cui si dispone; gli autori hanno infatti previsto la versione per dischetto e per cassetta.

Infine un foglio elettronico (spreadsheet), che permette di gestire, in modo automatico, calcoli complessi, analisi di mercato, budget e previsioni.

Un libro indirizzato a tutti, che consente di apprezzare il proprio computer non solo come console per videogame, ma anche come indispensabile strumento di lavoro.

input.output

**Piccoli
annunci**

Sono interessato all'acquisto di un disk drive per C 64, usato, ma in perfette condizioni di funzionamento. Scrivere o telefonare ore pasti a: Emanuele Silvestri - Piazza Malatesta, 2 - 47040 Verucchio (FO) - Tel. 0541/668040

Vendo, cambio oltre 3.000 validi programmi (utility, gestioni, giochi) in linguaggio macchina. Annuncio sempre valido. Telefonare ore pasti. Nazario Melchionda - Viale Teocrito, 108 - 96100 Siracusa - Tel. 0931/39898

Vendo VIC 20 con imballaggi originali completo di istruzioni, cavetti e trasformatore a L. 100.000; registratore Commodore a L. 60.000; 10 cartridge tra le quali Scacchi e Gorf a L. 5.000 cadauna; 4 cassette Mastertronics a L. 5.000 cadauna; una cassetta con due giochi a L. 3.000; 1 cassetta con 30 giochi completa di libro a L. 10.000; una cassetta Super Arcade con 10 giochi a L. 7.000. Il tutto, con pochi mesi di vita, in blocco al prezzo affare di L. 250.000 trattabili. Emanuele Filiberto - Viale Primo Maggio, 45 - 88046 Lamezia Terme (CZ)

Vendo o cambio programmi di giochi ed utility su disco per C 64. Invio la lista dei programmi a chi ne fa richiesta. Scrivere oppure telefonare dalle ore 12.30 alle 15.00. Fabrizio Porrovecchio - Via Capparozzo, 8 - 90146 Palermo - Tel. 091/244045

Per C 16 vendo 20 giochi tra i quali: Pac Man, Skramble, Bombolo, Robin To The Rescue, Al Ladro, Space Wars, Towel Of Evil al prezzo di L. 55.000 più spese postali oppure giochi a scelta. Sono disposto anche allo scambio. Fernando Taddei - Via C.D.A. Braida - 85014 Laurenzana (PZ) - Tel. 0971/961306

Cambio circa 200 programmi per C 64 fra giochi ed utility. Mandate le vostre liste a: Paolo Schirra - Via Taranto, 4 - 09170 Oristano - Tel. 0783/259666

Vendo software per C 64 a prezzi stracciati, da L. 3.000 a L. 10.000. Dispongo anche di Easy Script, Manager, Simons' BASIC ed Extended BASIC, che vendo a L. 15.000 cadauno. Richiedere la lista a: Renato Maniscalco - Via Duca Abruzzi - 91018 Salemi (TP)

Cerchi programmi di qualità per C 64? Io li ho; chiedi il catalogo a: Leonardo Tambone - Via Marconi, 47 - 20010 Cornaredo (MI) - Tel. 02/9319108

Compro, vendo, cambio software per C 16: prezzi ragionevoli, massima serietà, ottima qualità; tratto preferibilmente con la provincia di Livorno. Mandate la vostra lista a: Giuseppe Trinchini - Via S.Francesco, 20 - 57025 Piombino (LI) - Tel. 0565/34779

Vendo o cambio ultime novità per C 64: Two On Two, Summer Games II (cassetta), partita doppia, Gi Joe e altri. Telefonare dalle ore 16.00 alle 17.00. Gianmarco Marocchi - Via Roma, 66 - 38060 Tenno (TN) - Tel. 0464/500635

Vendo VIC 20 più 2 cartucce, 4 cassette con 50 programmi di utility e giochi, manuale, registratore, joystick ed alimentatore a L. 300.000 trattabili. Telefonare a: Alberto Bellina - Via L. Da Vinci, 518 - 90100 Palermo - Tel. 091/222922

Vendo, possibilmente al mio domicilio, C 64 più registratore, joystick, 4 cassette e 2 libri a L. 500.000. Ancora in garanzia. Telefonare dalle 19.00 alle 20.00. Alberto Marchetti - Via G. Leopardi, 7 - 24034 Cisano Bergamasco (BG) - Tel. 035/783284

Dispongo delle migliori novità per C 64 tra cui Exploding Fist (un karatè migliore di quello delle sale giochi), Karateka (un altro stupendo karatè), Rocky, Sky Fox, Two On Two (pallavolo), Beach Head II e moltissime altre. Notevole quantità di software anche su cassetta. Chiedere di Fabio. Fa.No. Software - Via Pestalozzi, 8 - 05100 Terni - Tel. 0744/88593

Vendo a L. 10.000 cassetta con 30 giochi per C 64: Turbo Dig Dag, Popeye, Fort Apocalypse, Wimbledon, Jungle Hunt, Defender, Pitfall, Basket, Super Pipeline, Suicide Strike, Aztec, Blue Moon, Grid Trap, Star Trek, One On One, Doctor 64, ecc. Davide Nebuloni - Via Piemonte, 47 - 20015 Parabiago (MI)

Eccezionale udite, udite o possessori di C 64: vendo Superturbo, fantastico programma che vi permette di "super-turbare" i vostri programmi, ma (cosa eccezionale) non richiede del suo caricamento nella ricerca. Il prezzo, strepitoso, è di L. 40.000. Per informazioni scrivere o telefonare a: Ettore Aquila - Via degli Stadi, 15 - 87100 Cosenza - Tel. 0984/393595

Vendo cassetta con oltre 10 programmi di giochi ed utility per C 16 e Plus/4 a L. 10.000; Cassetta per la riduzione dei sistemi Totocalcio a L. 6.000. Vendo inoltre, a L. 600.000, Plus/4 più alimentatore, corso BASIC con cassette, registratore e vari programmi. Un solo mese di vita. Guido Ricci - Via Capo Le Case - 67015 Montereale (AQ) - Tel. 0862/90242

Compro espansione 8-16 Kbyte per VIC 20 a prezzo conveniente. Telefonare a: Antonio Politi - Via Roma, 6 - 73015 Salice Salentino (LE) - Tel. 0832/731185

Sviluppo software di tipo gestionale su C 64 o Apple II. Tratto solo con Torino. Vincenzo Garrone - Via Mazzini, 47 - 10123 Torino - Tel. 011/833240

Cerco istruzioni del programma Simons' BASIC a prezzo ragionevole e possibilmente in italiano. In cambio sono disposto anche ad offrire giochi. Gianni Pierini - Via Dante, 403 - 52010 Capolona (AR) - Tel. 0575/48418

Vendo VIC 20 più 14 cartridge Commodore, manuale con istruzioni in italiano e 13 cassette. Inoltre regalo all'acquirente molte riviste un libro giochi. Tutto in ottime condizioni, L. 320.000. Romano Baglioni - Via Mergozzo, 13 (già E. Bondi, 107) - 00166 Roma - Tel. 06/6242159

Per C 64 vendo a L. 20.000 cassetta con 15 programmi a scelta. Annuncio sempre valido, massima serietà. Mauro Boiano - Via Del Giudice, 23 - 81010 S.Gregorio Matese (CE)

Cambio software per C 64. Dispongo di circa 150 programmi, tra cui parecchi giochi: Calcio, Pole Position, Rollerball, Popeye, Pitfall r. Mephisto. Fabio Fattini - Via Ruffio, 2558 - 47023 Cesena (FO) - Tel. 0547/52635

Ho fondato un club per i possessori di C 64. Chi si iscrive potrà acquistare videogame a prezzi bassissimi: per esempio, Ghostbusters a sole L. 3.500. Invio lista a chi ne fa richiesta. Tratto solo con la regione Veneto. Fabio Menti - Via G. Giardino, 6 - 36070 Valdarno (VI) - Tel. 0445/403088

Per C 64 vendo o cambio programmi di ingegneria su cassetta (adatti per studenti) con allegata spiegazione e Turbo Tape in regalo. Prezzi da L. 6.000. Francesco Pinizzotto - Via Tre Monti, 57 - 98100 Messina - Tel. 090/362011

Ehi, tu che leggi questo annuncio hai giochi (belli) per C64? Se vuoi scambiarli o venderli, hai trovato l'annuncio che fa per te. Puoi comunicare con me tramite tre canali di Input/Output, che sono: Lettura, Telefono, Modem. Questi tre canali sono a tua disposizione 24 ore su 24. Allora, cosa aspetti? Antonio Bianco - Via Montalbino, 16 - 84016 Pagani (SA) - Tel. 081/919350

Vendo Commodore 8032, espansione a 96 Kbyte, dual floppy Commodore 4040, stampante Commodore 3022, insieme o separatamente. Telefonare ore serali. Antonio Palumbo - Viale Elena, 26 - 80122 Napoli - Tel. 081/660068

Vendo software, riviste e libri nuovi su C 64, Spectrum e VIC 20 a prezzi molto convenienti. Giorgio Sasdelli - Via Vespri Siciliani, 31 - 20146 Milano - Tel. 02/4238485

Cambio o vendo software per VIC 20; dispongo di oltre 100 programmi. Spedite la vostra lista richiedendo la mia. Gaetano Scibilia - Via Duilio, 16 - 97019 Vittoria (RG) - Tel. 0932/988194

Vendo 500 splendidi programmi (sia su disco che su nastro) per C 64 a prezzi veramente ottimi. Chiedete la lista. Enza Albano - Via Verdi, 37 - 24040 Arzene (BG) - Tel. 035/878594

Vendo insieme o separatamente console Intellivision e 12 giochi: Calcio, Tennis, Basket, Boxe, Battaglia Navale, Automobilismo, Star Strike, Astro-smash, Space Armada, Poker, Ippica. Tutto allo stupendo prezzo di L. 250.000. Telefonare possibilmente ore pasti. Giò Esposito - Via Fratelli Kennedy, 2 - 20082 Noviglio (MI) - Tel. 02/9054330

Vendo VIC 20 più registratore, libro "Alla scoperta del VIC 20", 4 cassette con molti giochi in LM e 6 programmi di utilità a sole L. 300.000. Regalo un joystick all'acquirente. Scrivere o telefonare a: Orlando Stasi - Via dei Carpentieri, 3 - 80059 Torre del Greco (NA) - Tel. 081/8819608

Cambio programmi di ogni genere su disco o cassetta. Rispondo a tutti quelli che mi invieranno la propria lista di programmi. Annuncio sempre valido. Walter Bonato - Via S.Pio X, 1 - 31030 Borso del Grappa (TV)

Vendo VIC 20 più trasformatore, cavetti antenna, 3 cartucce giochi e una cassetta programmi. Tutto in ottime condizioni, a L. 200.000 trattabili. Telefonare ore pasti. Lorenzo Canepa - Via Turno, 7 - 80072 Arco Felice (NA) - Tel. 081/8667406

Eccezionale: vendo Turbo Tape per C 64 a 5 giri di nastro. Sì, avete letto bene: 5 giri. Lo vendo a L. 20.000, escluse le spese di spedizione ed il supporto. Giovanni Stefanelli - Via Badino, 206 - 04019 Terracina (LT) - Tel. 0773/731170

Vendo Summer Games II, Beach Head II, Two On Two, Rocky, Hot Wheels ed altri 3.500 programmi, tutti perfettamente funzionanti. Armando Mannino - Via Caronda, 196 - 95100 Catania - Tel. 447974

Vendo giochi per C 64 a L. 2.000 l'uno. Vendo cassetta con 20 giochi, tra i quali Frogger, One On One, Wimbledon e moltissimi altri, a L. 30.000. Giovanni Stefanelli - Via Badino, 206 - 04019 Terracina (LT) - Tel. 0773/731170

Vendo le seguenti cassette per Atari VCS 2600 a metà prezzo: Pac Man, Raiders Of The Lost Arc, Spider Man, Space Shuttle, The Activision Decathlon. Cristiano Cravanzola - Via Prof. Oliva, 16 - 12011 Borgo S.Dalmazzo (CN) - Tel. 0171/769659

Vendo a L. 20.000 cassetta contenente 11 programmi (tra giochi e utility) tra cui: Simons' BASIC, Speadsript, SAM, Calcio, ecc. Fabio Ferrera - Via dei Giustiani, 9 - 16036 Recco (GE) - Tel. 0185/73163

Sensazionale! Vendo giochi per C 64 (Frogger, Galaxia, King Kong, Skramble, ecc.) a L. 1.500 l'uno. Ne possiedo 60, tutti in Turbo Tape. Inoltre vendo anche utility (Turbo Tape, Archivio, ecc.) a L. 2.000 l'una. Giuseppe Antonini - Via Sempione, 25 - 28037 Domo-dossola (NO)

SCALDA IL JOYSTICK E GASATI CON



**SONO
IN EDICOLA**



**GRUPPO
EDITORIALE
JACKSON**

Milano-Londra-Madrid-San Francisco

IL VERO GIOCO COMINCIA ADESSO

Tascabili JACKSON



I piccoli, grandi manuali
pratici di informatica



GRUPPO EDITORIALE
JACKSON
DIVISIONE LIBRI

**NELLE MIGLIORI
LIBRERIE**