

SUPER

n° 3 Settembre 1984

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

3

VIC & C64

SUPERVIC
la rivista per gli utenti
dei prodotti Commodore

Una pubblicazione della

J.soft EDITRICE

in collaborazione con

GRUPPO
EDITORIALE
JACKSON



SUPERBASIC

DUELLO SPAZIALE

SIGNORE DEI DRAGHI

CYCLON ZAP

C64 SOUND TESTER

GP700A

stampante grafica a colori

SEIKOSHA



Dotata di un esclusivo sistema di stampa ad impatto a quattro martelli elaborato da Seikosha, la GP700A stampa quattro colori in simultanea e consente una grafica ad alta risoluzione in sette colori. I colori, definibili punto per punto, sono sovrapponibili senza problemi di interferenza sino ad ottenere un numero praticamente illimitato di sfumature di cui una sessantina ben apprezzabili.

La stampa avviene in maniera molto silenziosa grazie ad una custodia notevolmente insonorizzata.

La velocità di stampa di 50 CPS in ogni condizione d'uso è dovuta alla capacità di stampare più colori senza spostamenti del nastro.

È incorporata la funzione di autotest.

Caratteristiche:

- Stampante ad impatto a matrice di punti da 80 colonne
- Matrice di stampa 5x8 (densità 13 cpi) e 7x8 (densità 10 cpi)
- Percorso di stampa monodirezionale (da sinistra a destra)
- Capacità grafiche con indirizzamento del singolo dot
- Possibilità di ripetizione automatica di un carattere grafico
- Velocità 50 caratteri/secondo (matrice 5x8) e 38 caratteri/secondo (matrice 7x8)
- Caratterizzazione: 10 e 13 cpi e relativi espansi
- Interfacce: parallela centronics (optionals: Spectrum, Sinclair ZX81, seriale RS232C, seriale TTY 20 mA current loop, seriale TTL, seriale RS232C spooler, Commodore VIC 20 e 64, Apple II, IEEE 488 per CBM Commodore, video composita per Commodore VIC 20 e C 64)
- Alimentazione carta: trattori (larghezza modulo continuo variabile fino a 10") e frizione (foglio singolo)
- Stampa 1 originale e 2 copie
- Set di 116 caratteri ASCII
- Consumo 10W (standby) o 25W (stampa)
- Livello di rumore: inferiore a 60 dB
- Durata di vita testa: 80 milioni di caratteri
- MTBF: 2500 ore (escluso vita testa)
- Peso 6 KG
- Dimensioni: 320 (prof) x 450 (largh) x 113 (alt) mm; 395 (prof) mm. con separatore carta
- Nastro: a quattro colori su cartuccia dedicata.

SUPER

VIC & C64

- 7** EDITORIALE
POINT
SUPERVIC E C64 CRESCE!
di Pietro dell'Orco
- 9** **READ & WRITE**
POSTA
- 12** **OPEN**
MUSIC MACHINE, LA MUSICA ELETTRONICA A PORTATA DI TUTTI I MANS: SUL FILO DEI 300 CON IL COMMODORE 64
a cura della Redazione
- 14** **...then**
INIZIARE CON UN SISTEMA A DISCHI
di C. Brannon
trad. e adatt. di M. Cristuib Grizzi
- 16** **REM:HW**
AGGIUNGERE PERIFERICHE
di O. Cowper
trad. e adatt. di M. Cristuib Grizzi
- 23** **LOAD**
23 SUPER BASIC
di M.C. Kees
trad. e adatt. di N. Mandelli, F. Palazzi, O Calvi
- 36** **DUELLO SPAZIALE**
di A. Hayes
trad. e adatt. di F. Stella
- 41** **IL SIGNORE DEI DRAGHI**
di D. Berdon
trad. e adatt. di U.G. Barzaghi
- 48** **LISTA VARIABILI**
di E.A. Cottrellx
trad. e adatt. di U.G. Barzaghi
- 50** **CYLON ZAP**
di M. Dodley
trad. e adatt. di U.G. Barzaghi
- 60** **SHUTTLE IN FUGA**
di M: Giwer
trad. e adatt. di U.G. Barzaghi e R. Contini
- 70** **COME AUMENTARE LA VELOCITÀ DEL VIC**
di D. Carmichael
trad. e adatt. di U.G. Barzaghi
- 72** **DOTT. VIDEO 64**
di D.W. Martin
trad. e adatt. di U.G. Barzaghi
- 73** **VICsound**
C64 SOUND TESTER
di R.V. Picard
trad. e adatt. di M. Cristuib Grizzi
- 76** **BORSA PER C64**
di M. Cristuib Grizzi
- 78** **PRINT**
RECENSIONI LIBRI
di M. Cristuib Grizzi
- 80** **input.output**
PICCOLI ANNUNCI

J.soft s.r.l.
DIREZIONE, REDAZIONE,
AMMINISTRAZIONE
Via Rosellini, 12
20124 MILANO
Tel. (02) 6888228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE
Mauro Cristuib Grizzi

HANNO COLLABORATO A QUESTO NUMERO
U. Barzaghi
O. Calvi
P. Ferrari
F. Palazzi
F. Stella

GRAFICA E IMPAGINAZIONE
Margherita La Noce
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
d&b Via Vignola, 5
Tel. 02/59.85.08
20133 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionario per l'Italia e
l'Estero Reina s.r.l.
Via Washington, 50
20046 MILANO
Tel. (02) 4988066 (5 linee R.A.)
Tlx. 316213 REINA I

Concessionario esclusivo per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo (11 numeri)
L. 75.000; per l'estero L. 110.000
I versamenti devono essere
indirizzati a:
J.soft s.r.l.
Via Rosellini, 12
20124 MILANO

mediante emissione di assegno
bancario, cartolina, vaglia o
utilizzando il c/c postale
n. 19445204
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli

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

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA COMPUTE!
O DA COMPUTE! PUBLICATION, INC.
P.O.
BOX 5406,
GREENSBORO, NC 27403 - USA

Per collaborare a SUPERVIC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Commodore. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente "guida" permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

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

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 sottolineato dell'articolo 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 tutte le linee del testo dell'articolo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

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, ed anche su entrambi i lati nel caso si usino cassette. 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 suggerimenti di pro-

grammazione, 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, è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT, etc.). 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 fotografie, queste dovranno essere in formato 24 x 36 o 6 x 6, in bianco e nero o diapositive.

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

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. 50.000 ad un massimo di L. 300.000). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

Spedite i vostri lavori a:

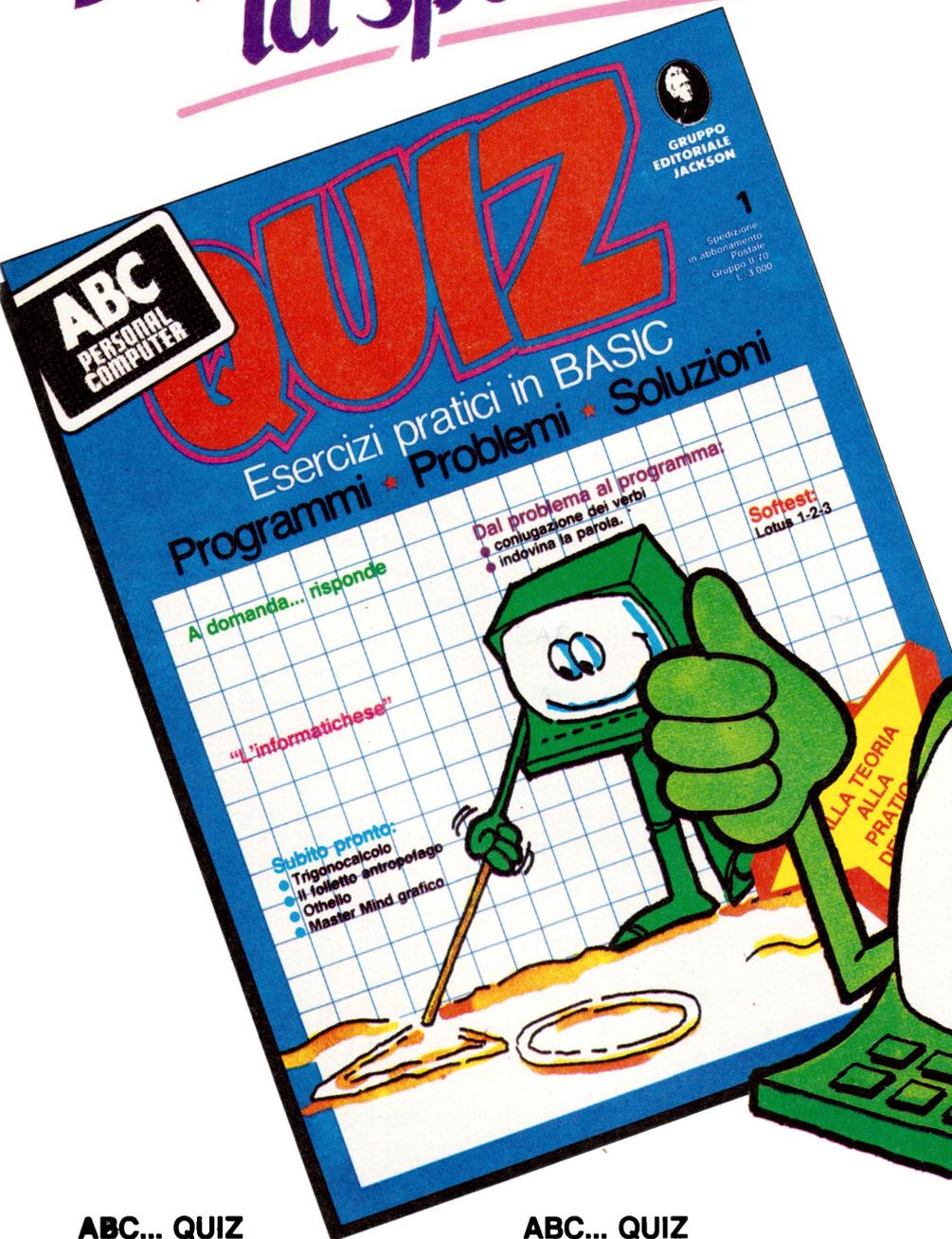
SUPERVIC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

Dopo il "corso" la "specializzazione"!

Dal 21 settembre
in edicola
**ABC
EDITORIALE
JACKSON**
QUIZ



ABC... QUIZ in 10 fascicoli

esercizi pratici, programmi, problemi, soluzioni per chi conosce il BASIC e vuole applicarlo

ABC... QUIZ in 280 pagine

i problemi, i loro algoritmi, la loro soluzione; il tracciamento dei diagrammi a blocchi; i listati dei programmi e le varianti a seconda dei diversi "dialetti" BASIC; i programmi professionali di largo uso: spreadsheet o fogli elettronici, word processor, data base...

ABC... QUIZ

1 volume (Formato cm. 21x 28)
ricco di illustrazioni, flowchart e programmi di gioco, utilità e pratica

ABC... QUIZ

un'opera da rilegare e custodire in biblioteca, è firmata



**GRUPPO
EDITORIALE
JACKSON**



Piccola guida per l'input dei programmi

Molti dei programmi listati da SUPERVIC contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, etc.).

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 per VIC 20 e C64 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 premuto insieme al tasto SHIFT, nel listato il simbolo di quel 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 «Commodore» 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 sul C64 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 predispose appunto in «modo virgolette».

Battendo un carattere qualsiasi cercando poi di correggerlo muovendo il cursore a sinistra, 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 spazi nella linea.

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

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>]	CTRL 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	CTRL 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>]	CTRL 1		{F4}	F4	
{RVS}	CTRL 9		[<2>]	CTRL 2		{F5}	F5	
{OFF}	CTRL 0		[<3>]	CTRL 3		{F6}	F6	
{BLK}	CTRL 1		[<4>]	CTRL 4		{F7}	F7	
{WHT}	CTRL 2		[<5>]	CTRL 5		{F8}	F8	
{RED}	CTRL 3		[<6>]	CTRL 6				

POINT

Supervic & C64 cresce!

La rivista e la cassetta per meglio rispondere alle esigenze dei lettori!

Caro lettore, nell'ultimo numero di SuperVIC & C64 siamo usciti con un editoriale in "difesa" della carta, rispetto al solo supporto magnetico.

Sono stati in molti a scrivere e a telefonarci segnalandoci che, se la carta è un mezzo insostituibile per verificare i programmi digitati e per tutte le rubriche e gli articoli sull'hardware, la cassetta è senz'altro più "comoda": per archiviare i programmi, per utilizzarli immediatamente, per "vederli" sul video da subito.

Il risultato è quindi un numero speciale: SuperVIC & C64 con cassetta allegata.

Certo, qualcosa è cambiato: il prezzo ad esempio, che peraltro abbiamo cercato di contenere, per non penalizzare troppo i lettori!

A tale riguardo dobbiamo considerare che esistono sul mercato, ad un prezzo superiore, prodotti che di "cartaceo" hanno ben poco, essendo il contenuto delle riviste limitato al solo indice dei programmi.

Siete soddisfatti della soluzione proposta o siete di diverso parere? Fatecelo sapere!

SuperVIC & C64 di settembre è un numero di "prova": siete Voi che dovrete comunicare, telefonando o inviando il questionario pubblicato qui sotto, quale soluzione preferite per rendere la rivista sempre più Vostra e rispondente alle Vostre esigenze.

In altre parole, preferite che la cassetta con i programmi sia sempre allegata a SuperVIC & C64, o desiderate che rivista e cassetta siano vendute separatamente, ad un prezzo complessivo ovviamente superiore?

Grazie a tutti per la collaborazione.

Il Direttore

Tagliando da compilare ed inviare in busta chiusa a: J.soft - via Rosellini, 12 - 20124 Milano

SUPERVIC & C64: LA PAROLA AI LETTORI

- Ottima idea: desidero che SuperVIC & C64 contenga in ogni numero la cassetta allegata.
- Pessima idea: preferisco vedere i prodotti in edicola separatamente, anche se in questo modo la somma dei due prezzi (rivista e cassetta) sarebbe evidentemente superiore.

Nome _____

Cognome _____

Indirizzo _____

cap. _____

Città _____

La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione. Un computer possiede potenzialità ma, come una macchina senza benzina, senza un programma, non può funzionare.

La maggior parte dei programmi pubblicati su **SUPERVIC** per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese, **SUPERVIC** pubblica programmi sia per il VIC che per il 64. Tanto per cominciare, se possiedi un VIC copia solo i programmi scritti per la tua macchina indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire certi programmi per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa.

Ogni lettera, carattere, o numero ha il suo significato.

Un errore banale è costituito dalla sostituzione nel numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere maiuscolo "B" con il numero "8".

Devi anche inserire tutta la punteggiatura, i due punti (;) e le virgole copiando esattamente ciò che appare sulla rivista. Gli spazi possono essere importanti. Per essere sicuro, copia il listato esattamente come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerete indicazioni tra parentesi quali: "{GIÙ}".

Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fate riferimento alla "Piccola guida per l'input" dei programmi".

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le necessarie informazioni di cui il programma ha bisogno.

In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, altri contengono codici grafici. Queste

linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato il calcolatore potrebbe "piantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.

Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria per cui è necessario sempre fare il SAVE del vostro programma prima di comandare il RUN.

Se il computer si ferma puoi caricare (LOAD) il programma e cercare l'errore.

A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.

Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.

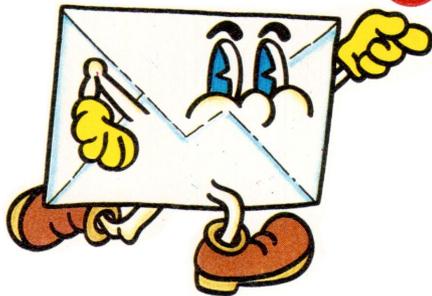
Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inverse", i caratteri minuscoli e quelli di controllo? Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

- 1) Copiare il programma una linea alla volta, con ordine. Premere RETURN alla fine di ogni linea. Usare il tasto "DEL" per correggere gli errori
- 2) Confrontare la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presenti un errore quando esegui il RUN del programma.
- 3) Accertati di avere inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input" dei programmi" che trovi in questa stessa rivista).

Siamo spiacenti di non poter rispondere alle singole richieste di informazioni circa i programmi, prodotti, o i servizi apparsi su SUPERVIC.

READ & WRITE



La posta
dei lettori

A proposito dell'OK-LIST

Sono un appassionato di computer e da sei mesi posseggo un VIC 20 con registratore a cassette. Ho trovato molto interessante il programma OK-LIST pubblicato su SUPERVIC, in quest'ultimo si possono correggere direttamente gli sbagli nella copiatura dei listati. Vorrei chiedervi se esiste qualche modo per re-inserire l'OK-LIST dopo averlo disattivato con i tasti RUN/STOP + RESTORE e dopo aver caricato in memoria un altro listato da correggere. Siccome per motivi di lavoro non riesco mai a terminare la copiatura di listato in una volta sola, sarebbe molto utile poter usare OK-LIST per controllare i vari pezzi di programma che battuto di volta in volta.

Angelo Saddi

R. Ci sono giunte moltissime lettere sul tenore di questa a proposito dell'uso di OK-LIST. Ricapitoliamo quindi i termini del problema: OK-LIST è un programma in linguaggio macchina che viene "pokato" nel buffer del registratore a cassette. Il buffer è una particolare area di memoria usata per immagazzinare temporaneamente i dati in passaggio tra computer e registratore e viceversa. Quando si esegue un LOAD o un SAVE, il passaggio dei dati nella stessa area in cui lavora OK-LIST provoca la cancellazione di quest'ultimo od addirittura il blocco del computer se OK-LIST non è stato prima disabilitato con RUN/STOP + RESTORE. È quindi evidente che OK-LIST può essere usato solo in fase di prima digitazione di un listato, a meno di disporre di un disk drive, nel qual caso non si hanno queste limitazioni. La scelta del buffer del registratore come area di memoria nella quale allocare OK-LIST non è certo casuale, ma nasce dall'esigenza di standardizzare il programma per l'uso sia con il VIC 20

che con il C64 e di fare in modo che questo non venga cancellato dal programma BASIC che si sta battendo.

Nel VIC 20 l'unica area di memoria con questi requisiti è appunto il buffer del registratore, mentre nel C64 si ha a disposizione anche l'area che inizia a 49152 decimale, e non è accessibile dal BASIC. Per aiutare almeno gli utenti di C64, abbiamo rilocato OK-LIST in questa zona di memoria. Questa versione di OK-LIST si attiva con SYS 49152 e si disattiva sempre con la pressione contemporanea dei tasti RUN/STOP + RESTORE. Rimane sempre l'obbligo di disattivare OK-LIST prima di un SAVE o LOAD, ma poi lo si potrà tranquillamente riattivare con la SYS di cui sopra.

Purtroppo, gli utenti di VIC 20, a causa della limitata quantità di memoria disponibile, dovranno accontentarsi della solita versione di OK-LIST allocata nel buffer del registratore.

Diamo qui di seguito il listato della variante di OK-LIST per il C64, sperando di avere accontentato molti dei lettori che ci hanno scritto.

```
100 PRINT"{CLR}ATTENDERE PREGO...":FOR I=4
9152 TO 49284:READ A:CK=CK+A:POKE I,A:NEXT
110 IF CK<>18178 THEN PRINT"{GIU'}ERRORE NELLE DATA":END
120 PRINT"{CLR}{ 2 GIU'}OK-LIST ATTIVATO":SYS 49152:NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,33,192,173
898 DATA 037,003,141,34,192,169
904 DATA 32,141,036,003,169,192
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,133
958 DATA 192,206,133,192,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 33,192,192,087,208,006
994 DATA 032,205,189,076,117,192
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 133,192,133,214,076,55,192
```

Cassette con il Music Editor

Complimentandovi con voi per la nuova rivista, colgo l'occasione per chiedervi, dal momento che mi sono avvicinato da poco al mondo dei computers (posiedo un C64), come trasformare il programma Music Editor per C64 in modo da poterlo utilizzare con il registratore a cassette e non con il floppy disk, ed inoltre se è possibile caricare due diversi programmi già registrati e come.

Gaetano Di Chiara

R. Il programma Music Editor per C64 nasce per essere utilizzato con floppy disk, tuttavia, come qualunque altro programma può essere modificato a seconda delle particolari esigenze di ciascuno. In particolare, la modifica per permettere l'utilizzo del registratore a cassette invece del floppy disk è concettualmente semplice, ma articolata da eseguirsi.

Le linee che contengono le istruzioni di apertura di un canale verso il drive sono la 2010 e la 3010, che vanno così modificate:

```
2010 GOSUB 760: OPEN 2,1,1,BR$
```

```
3010 GOSUB 760: OPEN 2,1,0,BR$
```

Con queste modifiche il programma caricherà e leggerà i dati sul registratore a cassette invece che sul drive. Rimane però il problema della comparsa delle scritte "Press play on tape" e "Press Record&Play on Tape" che provocano uno scroll verso l'alto di tutta la videata. Si può tuttavia avviare facilmente a questo inconveniente ponendo prima dell'istruzione dell'apertura del canale degli opportuni comandi di cursore in modo da provocare la comparsa delle scritte nella zona dello schermo che si ritiene più opportuna. Passando al secondo quesito, purtroppo a differenza di altri computer anche più economici, né il VIC né il C64 possiedono l'istruzione MERGE che permette di fondere in un unico programma quello che si trova attualmente in RAM ed un altro registrato su un apparato esterno.

Tuttavia questa istruzione è disponibile in vari tool per il C64, o può essere simulata ricorrendo a brevi routines in linguaggio macchina. Ci ripromettiamo appena possibile di pubblicarne una.

J. SOFT: OBIETTIVO



Il mondo del software e dei servizi di informatica è difficile da fotografare. Una domanda che varia di continuo, prodotti nuovi giornalmente immessi sul mercato. Non solo. La realtà italiana è ancor più complessa, per caratteristiche uniche di lingua, cultura, organizzazione.

Un mercato difficile, con infinite variabili. Noi della J. Soft abbiamo una risposta a tutte queste istanze: anzi, tutte le risposte. Perché siamo un editore di software, di riviste e di libri su supporto magnetico e su carta.

In più, unici in Italia, distribuiamo su tutto il territorio nazionale, con una nostra rete di agenti, programmi nostri e della prima e più prestigiosa software-house indipendente del mondo: MICROSOFT.

J. Soft con una struttura snella affronta con dinamicità tutti gli aspetti di questo nuovo mercato: dall'informazione alla formazione, dalla vendita all'assistenza post-vendita.

Il nostro cliente è un amico, sia esso un dealer, un utente finale, un abbonato.

Sempre a sua disposizione, raggiungibili telefonicamente con un numero diretto (02-6888228), per aiutarlo a risolvere un problema, a chiarire un dubbio, a far tesoro di tutti i suoi suggerimenti.

J. Soft vuole distinguersi per una filosofia commerciale di una geometrica semplicità: offrire il meglio, nella forma più adatta, a un prezzo conveniente, con il miglior servizio.

SOFTWARE!



Scegli anche tu

 **J. soft** EDITRICE

Il soft-publisher amico

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



a cura della **Redazione**

Music Machine, la musica elettronica a portata di tutti.

Il Commodore 64 è il personal computer che possiede in assoluto le migliori capacità di sintesi sonora e musicale.

A questo compito è infatti dedicato un intero microprocessore parallelo a quello principale, il SID (Sound Interface Device) 6581, capace di gestire suoni polifonici su parecchie ottave e con forme d'onda variabili.

Proprio a causa di questa versatilità o potenza creare musica con il Commodore 64 da ambiente Basic comporta la stesura di programmi piuttosto complessi, con numerosi POKE, DATA e cicli di ritardo.

Se si vuole impiegare il Commodore 64 alla stregua di un semplice strumento musicale elettronico dalle notevoli possibilità, anche senza alcuna nozione di programmazione o di informatica in generale, ciò è possibile facendo uso di un nuovo programma della Commodore chiamato "Music Machine".

Il Music Machine Commodore risiede su cartuccia ROM, quindi tutto quello che bisogna fare per cominciare a premere il pulsante dell'accensione.

Sullo schermo compare un pentagramma e alcune scritte indicanti lo stato del sistema.

Come tastiera si impiegano le prime due file di tasti del calcolatore.

La prima fila comanda i diesis e i bemolli (tasti neri del pianoforte), la seconda fila le note naturali.



L'estensione sonora del Commodore 64 trasformato in sintetizzatore, copre 6 ottave (selezionabili tramite i tasticursore), una sola ottava in meno di un pianoforte da concerto.

Ad ogni tasto premuto viene visualizzata sul pentagramma la nota corrispondente, mentre il suono esce dall'altoparlante del televisore o del monitor.

Per ogni suono, oltre all'altezza, è possibile selezionare anche la forma d'onda (quadra, triangolare, a dente di sega, ecc.), la forma dell'involuppo ADSR (Attack, Decay, Sustain, Release) e porre anche gli effetti speciali (Naturale, Vibrato e "Glide").

Per esempio un suono generato tramite un'onda triangolare ha un timbro piuttosto dolce e ovattato, adatto per un accompagnamento, mentre lo stesso generato da un dente

di sega sarà molto incisivo e spiccherà sul sottofondo melodico.

Similmente selezionando il registro Decay si avrà una nota simile a quella generata dagli strumenti tradizionali, mentre il Sustain darà un sound inconfondibilmente elettronico.

Anche il Vibrato e il Glide contribuiscono a colorire la musica ottenuta, mediante l'ultimo in particolare risulta possibile ottenere l'effetto di "scivolata" passando da una nota all'altra, tipico delle migliori chitarre elettriche.

Mediante un apposito comando (tasto "Shift" di destra) si possono ottenere anche effetti polifonici e accordi.

I tasti di funzione sono in grado di fornire un accompagnamento ritmico.

Oltre a insegnare i primi rudimenti della teoria musicale e il riconoscimento delle note il Music Machine Commodore ha caratteristiche tali da poter essere impiegato a livello amatoriale come vero e proprio strumento musicale, tenendo soprattutto presente il basso costo del Commodore 64 rispetto a qualsiasi tipo di tastiera elettronica e il fatto che questo calcolatore è già predisposto per essere collegato con un amplificatore o un impianto hi-fi tramite un semplice cavetto DIN a cinque poli.

Music Machine Commodore è in vendita a L. 41.000 + IVA

LE MANS: sul filo dei 300 con il Commodore 64



Bastano una cartuccia-gioco e una paddle collegata al computer, ed ecco che i motori rombano impazienti, in attesa del segnale di via libera per gettarsi all'inseguimento degli avversari nel più famoso Gran Premio automobilistico di Formula Uno.

Con Le Mans il nuovo gioco realizzato dalla Commodore per il suo 64, i vertuti possono emulare le gesta di Niki Lauda al volante di una rombante Formula Uno, cimentandosi in una emozionante gara contro il tempo sul filo dei 300 orari. Destreggiandosi fra chicanes, curve ad esse, tratti ghiacciati e percorsi in notturna, bisogna cercare di tenere la media più alta, evitando di urtare gli avversari che caracollano sulla pista.

Lemans è dotato di una grafica e di un sonoro paragonabili a quelli dei più famosi videogames da bar, e riesce a esprimere al meglio le potenzialità del Commodore 64. L'obiettivo del gioco è quello di superare gli avversari mantenendo una velocità elevata senza

uscire di pista o scontrarsi con le altre auto in gara.

Lemans è facile da usare: basta infilare la cartuccia nella fessura posteriore del Commodore 64, collegare la paddle all'apposita presa e accendere il computer. Il gioco si carica da sé, e chiunque è in grado di battersi contro temibili avversari pronti a tutto pur di ostacolare la lunga marcia verso il traguardo.

La manopola della paddle si usa come volante, e ruotandola ci si sposta a destra e a sinistra lungo la corsia di marcia.

Il bottone di sparo funziona da acceleratore: tenendolo premuto si aumenta la velocità e si ottengono scatti brucianti. La strategia vincente è quella di guidare sempre "a tavoletta", senza mai scendere sotto i 300 Km/h, e di superare il maggior numero possibile di avversari. In caso di incidente bisogna raggiungere immediatamente i box evitando di perdere secondi preziosi nell'attraversamento della carreggiata.

La guida sul tratto di pista ghiacciata è piuttosto difficile, perché i movimenti lungo la carreggiata risultano amplificati dalla superficie scivolosa. Niente a che vedere comunque con la guida notturna, durante la quale la pista è illuminata soltanto dai fari della propria auto.

Chi riuscisse a superare anche le famose "Esses", le micidiali curve che hanno reso celebre il circuito di Le Mans, dovrà comunque imboccare uno strettissimo viadotto, ingombro di bolidi avversari ben felici di ostacolare il cammino.

Ma arrivare al traguardo non è comunque un problema, perché la gara termina sulla dirittura finale allo scadere del tempo di gioco.

Lemans è in vendita a L. 41.000. = + IVA

Sul prossimo numero troverete:



GRAFICA BITMAP SUL C64: finalmente svelati tutti i trucchetti per ottenere semplicemente dei disegni in alta risoluzione sul vostro C64.



DISEGNATORE: grafici e disegni tridimensionali resi automatici da questo articolato programma che lavora completamente in alta risoluzione.



TERAPIA: risolvete i problemi del vostro inconscio ricorrendo al vostro fido VIC o C64 che vi farà risparmiare il ricorso all'analista.



TANTI ALTRI programmi, giochi ed utilities per i vostri VIC e C64 assetati di SUPER!

Iniziare con un sistema a dischi

di C. Brannon
trad. e adatt. di M. Cristuib Grizzi

Parte prima

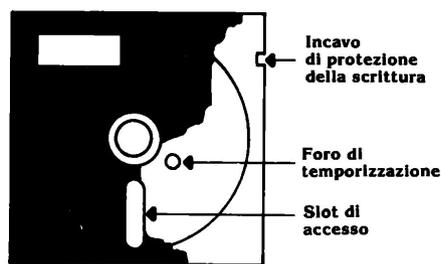
Un disk drive può rendere il caricamento ed il salvataggio dei programmi molto più semplice ed amplificare la velocità e la potenza delle vostre applicazioni. In ogni caso, un disk drive è molto di più di un registratore a cassette, e la sua notevole complessità può essere causa di malintesi; questa serie di articoli si ripromette di esplorare i vantaggi del sistema a dischi, approfondendo quanto riportato dal manuale dei drive VIC 1540 e VIC 1541, ed anche qualcosa di più.

Può darsi che siate ormai stanchi di cercare la posizione di un programma su una cassetta, e poi aspettare cinque o dieci minuti per caricare un programma un po' lungo... Oppure state usando con il registratore un bellissimo e completo programma di gestione dati per la vostra attività, e scoprite che forse fareste prima a trovare un record da voi piuttosto che tramite il computer... E che dire degli errori di caricamento che vi perseguitano? D'altra parte, il prezzo relativamente basso del disk drive Commodore potrebbe sembrare esorbitante se i vantaggi fossero solo velocità e facilità d'uso; e se non vi importa di attendere dieci minuti per caricare un gioco, probabilmente non avete bisogno di un disk drive. Se invece il registratore è l'anello più debole nella catena del vostro "sistema", otterrete un sostanziale innalzamento di prestazioni dall'investimento.

In ogni caso, questa serie di articoli vi renderà edotti su ogni vantaggio ed ogni pecca del sistema a dischi Commodore serie 1540 e 1541. Vi daremo le informazioni utili per iniziare ad usarlo come tutti i trucchetti per ricavarne le massime prestazioni.

Per prima cosa, vediamo come è fatto un disco (Figura 1): è come un sottile, flessibile (floppy) disco audio a 45 giri, con la differenza che è racchiuso da una busta di plastica. I floppy disk possono avere un diametro di 8 pollici o di 5 pollici e un quarto (minifloppy); que-

Figura 1 vista in sezione



sti ultimi sono i più diffusi per i personal computer. Recentemente alcune marche di computer hanno introdotto i microfloppy, ma non si è ancora deciso se la misura standard debba essere 3 pollici, o 3 ed un quarto oppure ancora 3 e mezzo. I dischi sono costituiti da mylar flessibile ricoperto dal familiare ossido magnetico del nastro da registrazione: è stato fatto molto lavoro per incrementare le prestazioni del floppy disk, ma la sua struttura basilare è rimasta sempre la stessa. Le grosse case produttrici di floppy disk (Scotch, Verbatim, BASF, Maxell, ed altre) ricavano i dischi da larghe piastre di mylar, e tutti i dischi - a singola e doppia faccia ed a singola e doppia densità (spiegheremo più avanti il significato di questi termini) - sono ricavati dalla stessa piastra. Quando passano al controllo di qualità, vengono classificati un po' come le uova, e non è molto consolante sapere che il disco medio e faccia singola e singola densità (cioè la classificazione più bassa) deve fallire numerosi test per entrare in questa categoria.

Fortunatamente per noi utenti, la sempre più elevata domanda di dischi a singola faccia/singola densità costringe le fabbriche a mettere in commercio con questa classificazione anche dischi di qualità superiore, al solo scopo di soddisfare le esigenze del

mercato. Quando inserite il disco nel drive, un apposito perno di quest'ultimo va ad inserirsi nel suo foro centrale, ma il centraggio non è mai perfetto. Ecco un primo trucco: per migliorare il centraggio inserite il disco mentre il motore sta girando (per farlo girare battete LOAD "\$", 8). La testina di registrazione e lettura del drive, che è simile a quella del registratore, deve avere un buon contatto con la superficie del disco: quando la fessura di ingresso è chiusa, il disco viene premuto dall'alto in modo che sia bene a contatto con la testina; ecco perché il disco deve essere flessibile. Essendo la testina sotto il disco, è su questa facciata che esso viene registrato: tenetene conto quando lo maneggiate. La testina mobile (al contrario di quella del registratore che è fissa) è montata su un binario rettilineo che le permette di muoversi avanti ed indietro attraverso la superficie del disco: un sistema abbastanza simile a quello usato sui giradischi stereo a tracciamento lineare. Infatti, l'analogia con il giradischi audio è molto utile per comprendere il funzionamento del disk drive.

Se il vostro giradischi avesse la velocità di rotazione del disk drive (300 giri al minuto) i vostri LP preferiti produrrebbero solo una sequenza di acutissimi stridii. Ciò può dare un'idea di quale sia la differenza di velocità con il registratore a cassette, il quale muove il nastro a 4.75 centimetri al secondo.

Certo la velocità di rotazione del disco conta molto ai fini della rapidità del caricamento di un programma, ma non è l'unico fattore che rende il disk drive così rapido; egualmente importante è il modo con il quale il drive trova ciò che sta cercando sul disco. Al contrario del registratore, il drive può muovere la sua testina di registrazione / lettura direttamente sulla parte del supporto magnetico nella quale sono immagazzinati i dati che ci servono. Con un registratore — sia quello dello ste-

reo che quello del computer — bisogna cercare ciò che vogliamo (sia una canzone che un programma) in modo sequenziale lungo il nastro; in altre parole, il nastro scorre davanti alla testina, che deve aspettare a caricare fino a che non le passi davanti la parte di nastro che cerchiamo. Questo sistema, chiamato "accesso sequenziale", è largamente inefficiente. Certamente potete premere il tasto di avanzamento veloce fino alla parte di nastro dove — pensate — inizi il programma (e non è mai lì...) oppure usare il contanastro per farlo giungere al numero che avete segnato su un foglietto (ma... che fine ha fatto?); inoltre il contanastro non è assolutamente preciso e le sue indicazioni variano enormemente tra registratori diversi pur dello stesso modello e marca. Su un giradischi, invece, non avete altro da fare che alzare il braccetto con la testina e porlo direttamente all'inizio del brano che volete ascoltare, saltando di getto tutti i precedenti: un bel risparmio di tempo, no? Il disk drive lavora proprio così: con il sistema ad "accesso casuale".

Ovviamente, le tracce del floppy sono molto diverse dai solchi di un LP: quest'ultimo possiede in realtà un unico solco che procede a spirale verso il centro, mentre le tracce del floppy sono tutti anelli concentrici. Ogni traccia è a sua volta suddivisa in parti chiamate "settori", ed ogni settore contiene un "blocco" di informazioni. La grandezza dei settori è variabile secondo la marca del drive, e nei Commodore è di 256 byte. Molti dischi posseggono 40 tracce, ed ogni traccia ha generalmente lo stesso numero di settori; un disco medio a singola densità può contenere circa 100K (100000 byte), ma ciò è variabile secondo il numero di settori per traccia. I dischi a doppia densità hanno un numero superiore di settori per traccia, cosa che richiede appunto una maggiore accuratezza di costruzione e qualità di materiale usato.

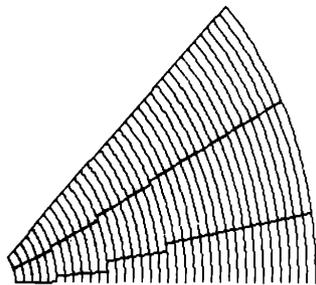
La Commodore ha risolto brillantemente la problematica della densità dei dischi: il disco per il drive 1540/1541 può contenere sino a 170Kbyte con una conformazione a singola densità. Ciò è reso possibile dall'uso di un numero variabile di settori per traccia (se si usano solo 35 tracce). Le tracce esterne di un disco sono più capaci e possono proporzionalmente contenere un numero di settori superiore rispetto alle tracce interne (vedi Figura 2); il sistema usato dalla Commodore risolve anche il problema lamentato da qualche tipo di drive quando si cerca di leggere o di scrivere sulle tracce più interne, dove i settori sono stipati in modo più severo.

A questo punto è opportuno introdurre i concetti di "hard sectoring" e di "soft sectoring". I drive hard sectoring richiedono dei dischi con una serie di fori che sono letti attraverso un foro di temporizzazione (vedi la Figura 1); i fo-

ri dicono al drive la posizione di ogni settore. I drive soft sectoring, invece, sono in grado di localizzare un dato settore in base al tempo di rotazione del disco. Quest'ultimo può variare dallo Standard di 300 giri al minuto, a seconda dell'accuratezza con la quale si sia posizionato il disco nel drive. Con questo sistema è sufficiente un solo foro di temporizzazione, anche se certi drive come gli Atari non ne hanno addirittura alcuno. I disk drive che usano il sistema di hard sectoring (fori multipli) non possono usare dischi soft sectoring, mentre si possono usare entrambi i tipi sul vostro drive soft sectoring 1540/1541.

Fortunatamente, non dovete tenere a mente tutte queste informazioni quando usate il drive, dal momento che il sistema operativo del disco (DOS: Disk Operating System) si preoccupa auto-

Figura 2 configurazione
Traccia/settore-1/8 di superficie



(I settori sono suddivisioni di tracce concentriche)

maticamente della gestione dei settori; tutto ciò che dovete fare è dire al computer se volete leggere o scrivere dei dati, ed il nome del file interessato. Inoltre, molti programmi applicativi compiono anche queste operazioni per voi: in BASIC ad esempio dovete solo dare i comandi SAVE o LOAD (che verranno approfonditi più tardi); un word processor vi chiede semplicemente il nome del file e si prende cura di tutte le operazioni attinenti al salvataggio o caricamento del file stesso. In ogni caso sarete responsabili del buon mantenimento dello stato fisico dei vostri dischi; a questo scopo sarà opportuno seguire attentamente alcuni semplici regole in proposito.

Il disk drive rappresenta un investimento, e richiede quindi di essere trattato con cura per fornire sempre i migliori risultati. Tenete i vostri dischi ad una ragionevole temperatura dell'ambiente; per parafrasare un costruttore di dischi per computer, "Se la temperatura è confortevole per voi, lo sarà anche per i vostri dischi."

Maneggiate con cura i floppy disk: anche se a volte possono ancora funzionare dopo essere stati piegati in due (se siete molto fortunati), è meglio non prendere la parola "floppy" troppo letteralmente. Ogni piccola piegatura nel disco può causare una rottura del film superficiale magnetico e provocare degli errori di lettura.

Un'ultima raccomandazione è molto importante: dal momento che i dischi immagazzinano le informazioni in modo magnetico, teneteli ben lontani dai campi magnetici. La fonte più comune (e più dannosa) di campi magnetici in un sistema casalingo è l'apparecchio TV o il monitor; non appoggiate mai i dischi sul televisore e non sistemati mai a meno di trenta o quaranta centimetri di distanza dall'apparecchio. Sono in realtà molti gli apparecchi elettrici che generano campi magnetici pericolosi per i floppy disk: trasformatori, alimentatori, frigoriferi e refrigeratori, stampanti, etc. In generale ogni apparato elettrico genera un campo magnetico, debole (nella maggioranza dei casi, fortunatamente!) o forte che sia, quindi è sempre buona regola tenere i propri dischi il più lontano possibile da questo.

È a questo punto, credo, superfluo consigliare di tenere le calamite ad una distanza "chilometrica" dai vostri floppy...! Ed inoltre è meglio non sistemarli vicino ad un apparecchio telefonico: ogni volta che questo suona, genera un campo magnetico.

I produttori ci dicono di trattare i dischi come se fossero uova; tuttavia in realtà questi sono abbastanza resistenti a stress meccanici. Le persone che per lavoro hanno quotidianamente a che fare con i floppy amano raccontare storie stile "horror" su come dischi apparentemente irrecuperabili sono invece stati riutilizzati; la mia preferita è la seguente: un tool BASIC estremamente utile registrato su un disco incopiabile era rimasto vittima di un rovesciamento di bicchiere contenente sciropo di lampone. Potete facilmente immaginare lo stato del disco dopo due giorni quando me ne sono accorto con un colpo al cuore... Ebbene — ho pensato — perso per perso, si può tentare un trattamento estremo: estratto molto attentamente il disco dalla sua "veste" nera, lo ho lavato sotto acqua corrente con un detergente molto leggero. Dopo averlo lasciato asciugare tra due tovaglioli di carta, ho preso una nuova copertura da un disco nuovo e l'ho sfruttata per quello danneggiato, chiudendo l'apertura praticata per sfilarla con un po' di nastro adesivo. Il disco ora funzionava perfettamente senza alcun problema! In ogni caso, devo ammettere di essere stato molto fortunato...

Perché è così importante recuperare un disco danneggiato? La ragione è che un disco può contenere 170K, così se un floppy pieno si rovina, è veramente molto quello che si perde. Per questo è SEMPRE opportuno, quando è possibile, fare delle copie dei propri dischi, ed aggiornarle man mano che nuovi programmi si aggiungono ai precedenti. Il sistema della "Back-up copy" ci salverà sempre ed in ogni caso da eventualità come quella di cui sopra.

REM:HW

Hardware

Aggiungere periferiche

di O. Cowper
trad. e adatt. di M. Cristuib Grizzi

Costruire il "sistema" Home Computer

Non è escluso che una periferica possa costare più del vostro computer: così, fare il giusto acquisto è un dovere per il portafogli. Di quale periferica avete bisogno? Necessiterà di software a parte una volta acquistata? Questo articolo cerca di offrire una panoramica sull'argomento.

I fanatici del computer spesso si vantano dei propri "sistemi", creandovi attorno un'aura di sofisticatezza da centro di calcolo spaziale; questi sistemi spesso consistono in un microcomputer circondato da un più o meno vasto assortimento di periferiche.

Una periferica è in generale qualsiasi cosa si colleghi al computer; anche senza rendervi conto, avete iniziato ad aggiungere periferiche e costruire il vostro sistema al momento stesso dell'acquisto del computer: la prima periferica è sempre l'apparecchio TV od il monitor, e la seconda è in genere il registratore a cassette per caricare e salvare i programmi. Ed il joystick: non è forse anch'esso una periferica? Da qui in poi, l'elenco delle possibili periferiche per il vostro computer è piuttosto lungo: stampante, unità a dischi, modem, cartridge di espansione della memoria, penna ottica, sintetizzatore di voce, plotter, etc.

Tocca a voi decidere quali siano le periferiche delle quali avete bisogno: un disk drive per fare seriamente del word processing, un monitor a colori per immagini più definite, un modem per telecomunicazioni... Sarà la vostra cono-

scenza del settore a decidere la quantità di tempo spesa a comparare prezzo e qualità dei modelli della periferica scelta disponibili sul mercato. Se avete già provato a buttare un occhio nei negozi specializzati, vi sarete certamente accorti che non è affatto strano per un apparato periferico costare più del computer al quale è collegato.

Qual'è la scelta migliore?

Il modo più ovvio per garantirsi da problemi di interfacciamento tra periferiche e computer è quello di acquistare prodotti della stessa casa costruttrice del calcolatore stesso. Tuttavia non sempre questi offrono il miglior rapporto qualità-prezzo e spesso sono prodotti di altre aziende venduti con il marchio della casa costruttrice del computer: sovente le grosse compagnie acquistano l'hardware in elevate quantità presso altre industrie elettroniche e lo distribuiscono con il proprio marchio.

In molti di questi casi il prodotto è direttamente acquistabile "alla fonte" risparmiando una certa percentuale del prezzo d'acquisto. La chiave per scoprire la maggiore o minore convenienza di questa speculazione è appurare se è stato solo modificato il marchio "tout court", oppure se sono state fatte anche delle modifiche hardware per adattare la periferica ad un particolare computer: ad esempio, come tutti ben sappiamo, i computer Commodore possiedono un set di caratteri grafici direttamente accessibili da tastiera; la Commodore vende una stampante costruita da un'altra compagnia ma modificata per stampare questo set di ca-

ratteri. Stampanti del tutto simili sono disponibili anche di altre marche, ma non possono stampare i caratteri grafici del set Commodore se non acquistando a parte un chip ROM addizionale ed installandolo nella stampante.

Il prezzo non dovrebbe essere l'unico elemento di giudizio quando si tratta di decidere quale periferica acquistare: un altro fattore di primaria importanza da considerare è la facilità di interfacciamento. Usando ancora come esempio le stampanti, una stampante da 500.000 lire che richieda un modulo di interfaccia da 100.000 lire per essere collegata al vostro computer, non è preferibile ad un modello da 600.000 lire che sia invece direttamente collegabile. In aggiunta, non tutte le interfacce sono identiche: un amico ha acquistato di recente un'interfaccia che poteva essere usata per collegare un VIC 20 ad una serie di popolari stampanti a matrice di punti. Ingenuo, pensava di collegarla semplicemente al VIC da una parte ed alla stampante dall'altra, mentre solo dopo un'attenta lettura del manuale di istruzioni ha scoperto che si doveva aprire la stampante e saldare un cavo che provvedesse all'alimentazione dell'interfaccia! Una modifica del genere avrebbe sicuramente invalidato la garanzia della stampante, oltre a creare le notevoli perplessità che sempre sopraggiungono quando si devono mettere le mani nell'hardware. È proprio questo il tipo di dettagli che è bene investigare prima dell'acquisto di una periferica che non sia direttamente collegabile al vostro sistema. Inoltre bisogna sempre ricordarsi di investigare circa le necessità di speciali cavi di collegamento al

computer: non è raro il caso dell'acquirente che soddisfatto ritorna a casa dal negozio con la periferica, per scoprire la necessità di un connettore speciale per effettuare il collegamento. E poi magari ci si accorge che questo connettore costa cinquantamila lire. Gli utenti con esperienza di saldature possono costruirsi queste connessioni in proprio risparmiando il costo del cavo, ma non essendo questo un lavoro per principianti, è sempre opportuno controllare che con la periferica vengano forniti anche tutti i necessari connettori.

Il software

Un altro parametro da considerare è il supporto software richiesto: alcune periferiche non richiedono software aggiuntivo per il funzionamento, mentre altre, se ne sono prive, diventano completamente inservibili. Ad esempio, la stessa penna ottica può essere usata indifferentemente su un Atari, VIC o C64, ma senza un apposito software per leggere il valore corrispondente alla posizione della penna e convertirlo nella locazione video equivalente del vostro particolare computer, la penna non vi servirà a molto. Se non siete ancora programmatori esperti a sufficienza per scrivere in proprio il software necessario, assicuratevi sempre che questo venga fornito insieme alla periferica che state acquistando. Ciò vale soprattutto per apparati complessi come sintetizzatori vocali e plotter.

Mancanza di standardizzazione

Esistono pochi standard per le periferiche degli home computer. Molto di questa carenza di standardizzazione è dovuto al fatto che le diverse case produttrici seguono differenti filosofie di progetto. Ad esempio, il funzionamen-

to meccanico di tutti i drive per floppy disk da 5 pollici e 1/4 è essenzialmente identico, tuttavia quasi nessuna di queste unità è intercambiabile con le altre: usando lo stesso hardware di base, un drive Commodore immagazzina i dati sul disco in 683 settori di 256 byte ciascuno; un drive del TRS 80 Color Computer in 630 settori di 256 byte, un drive Apple II in 560 settori di 256 byte, ed un drive Texas Instruments in 360 settori di 256 byte. I drive Atari poi usano 720 settori, ma ogni settore è lungo solo 128 byte. Il Commodore scrive la directory sulla pista 18 del disco, l'Apple ed il TRS 80 sulla 17, ed il TI sulla pista 0. Tutto ciò non solo rende i drive incompatibili, ma significa anche che i dischi registrati su uno non possono essere letti su un altro di questi drive. Ogni costruttore ha solidi argomenti per sostenere che il proprio sistema è il migliore, e nessuno sembra voglia scendere ad un compromesso in nome della compatibilità.

Interfacce

Qualche tentativo di standardizzazione è però stato fatto. Ad esempio un'industria, la Centronics, era una delle principali fabbriche di stampanti per computer, ed usava uno schema di interfaccia parallela in cui i dati erano inviati alla stampante un byte alla volta. Le industrie che entrarono nel mercato in tempi successivi usarono la connessione Centronics, in modo che le loro stampanti potessero essere direttamente collegabili ai computer costruiti per le stampanti Centronics. Così questo sistema di connessione, con la sua spina a 36 poli, diventò di fatto uno standard, e le interfacce parallele Centronics sono ora disponibili per molti home computer. Uno standard più formale è stato invece definito per la trasmissione di dati in modo

seriale: lo standard, chiamato RS-232, consiste in un insieme di cavi che include una linea di trasmissione dati, una linea di ricezione dati, ed una collezione di linee di segnale "handshake" per regolare il trasferimento dei dati. Più in particolare, un connettore chiamato DB25 è quasi sempre usato sulle linee RS-232, ed a ogni polo di questo connettore è stato assegnato un particolare segnale della RS-232. Così, se il vostro computer possiede una porta RS-232, potete interfacciare senza problemi ogni periferica RS-232 compatibile, giusto?

Sbagliato. Sfortunatamente, lo standard RS-232 definisce un set di linee di segnale, ma si dimentica di stabilire quale deve essere considerato un segnale valido su queste linee: alcuni sistemi RS-232 usano come livelli di segnale di riferimento +12 volt e -12 volt, altri +5 V e -5 V, e pochi altri ancora usano +5 V e 0 V. In particolare il VIC 20 ed il C64 posseggono del software su ROM per la gestione delle comunicazioni RS-232 attraverso la User Port, ma se volete usare la maggior parte delle periferiche non Commodore, dovrete sistemare nella User Port stessa un modulo di interfaccia per aumentare i livelli di tensione in uscita. Dovete tener conto di questo se pensate di acquistare una periferica "standard" RS-232. Forse nel futuro qualche industria arriverà a dominare una fetta sufficientemente larga del mercato degli home computer per potere determinare uno standard generale. Alcuni dicono che sarà il PC IBM a diventare l'archetipo, altri confidano nell'esperimento nipponico-americano di standard MSX... per il momento "consoliamoci" pensando che per qualsiasi tipo e marca di computer si trova molto più equipaggiamento incompatibile di quanto se ne trovi di compatibile.

**Per ricevere la cassetta contenente
i programmi di SUPERVIC
compilate e spedite il coupon qui sotto!**

SUPERVIC

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Vogliate inviarmi n° cassette

contenenti programmi pubblicati sul numero..... di SUPERVIC al prezzo
speciale di L. 15.000 + 2.000 per spese di spedizione cadauna

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a J.soft - Milano

Pagherò in contrassegno al postino al ricevimento delle cassette

buono d'ordine per la cassetta contenente i programmi pubblicati sul numero... di SUPERVIC

Nome

Cognome

Via

Città C.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)

OK-List per la perfetta battitura dei listati

Inserite in memoria l'OK-LIST prima di accingervi a battere i listati pubblicati: avrete a disposizione un formidabile mezzo per essere sicuri di aver inserito nel computer ogni carattere in modo assolutamente corretto.

Tutti i nostri listati sono caratterizzati da un numero aggiunto ad ogni linea di programma, ad esempio: rem 123. NON COPIATE QUESTA PARTE DELLA LINEA! Serve unicamente per vostra informazione. L'istruzione REM ha l'unico scopo di renderla inoffensiva se per errore dovete batterla.

Se, prima di iniziare il lavoro di battitura del listato, caricate in memoria OK-LIST e lo attivate con SYS 886, per ogni linea che battete verrà visualizzato un numero (detto di checksum) nell'angolo superiore sinistro dello schermo.

Confrontate questo numero con quello pubblicato per ogni linea del listato: se non corrisponde, ciò significa che avete commesso un errore nella battitura della linea stessa.

Le ore buttate per cercare e correggere errori di battitura in programmi che non funzionano saranno solo un lontano ricordo!

Due ultime cose di cui tenere conto: OK-LIST non tiene conto degli spazi: questo per vostra convenienza, poiché questi generalmente sono poco importanti.

Infine, OK-LIST è allocato nel buffer del registratore (locazioni 886-1018), quindi prima di salvare su cassetta il programma che state battendo, ricordatevi di disabilitarlo premendo contemporaneamente RUN/STOP + RESTORE. Potrete poi riattivarlo con SYS 886.

Il sistema più comodo per usare OK-LIST è quello di battere normalmente il programma, quindi listarlo e portarsi con il cursore sul numero di linea più basso che appare sul video; premere quindi RETURN e confrontare il numero di checksum visualizzato nell'angolo superiore sinistro del video con quello pubblicato per la data riga del programma. Il cursore si porterà automaticamente sulla linea successiva, e non dovrete fare altro che premere nuovamente RETURN per ottenere il checksum seguente.

Data l'area di memoria in cui si trova, OK-LIST non può essere usato per controllare un programma già salvato su cassetta, mentre ciò è possibile se il programma è stato registrato su disco.



OK-LIST versione per VIC 20 e C 64

```

100 PRINT"{CLR}ATTENDERE PREGO..."
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:
NEXT
110 IF CK<>17539 THEN PRINT"{GIU'}ERRORE
NELLE ISTRUZIONI DATA":END
120 SYS886:PRINT"{CLR}{ 2 GIU'}OK-LIST AT
TIVATO.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003

```

MLX nuova versione

di C. Brannon

trad. e adatt. di F. Stella e

M. Cristuib Grizzi

Per l'introduzione senza errori di programmi in linguaggio macchina.

MLX è un programma che permette di inserire listati in linguaggio macchina in modo assolutamente sicuro 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 l'indirizzo della lo-

cazione 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 finea.

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

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. Il programma sarà mandato in esecuzione con una SYS (indirizzo di partenza).

Presentiamo in questo numero una versione migliorata di MLX per C64 pubblicato sullo scorso numero, unitamente ad una versione per VIC 20.

Chi avesse digitato la precedente versione può sostituire semplicemente le linee del programma che sono state modificate.

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-RIT
A 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"[<X>]"; :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

```

```

:rem 69
665 POKE254,S/256:POKE253,S-PEEK(254)*256
:POKE780,253 :rem 12
666 POKE782,E/256:POKE781,E-PEEK(782)*256
:SYS65496 :rem 124
670 IF(PEEK(783)AND1)OR(ST AND191)THEN780
:rem 111
675 PRINT"{GIU'}OPERAZIONE CONCLUSA.":END
:rem 152
680 PRINT"{GIU'}ERRORE DI REGISTRAZIO-NE.
RIPROVA.":IFDV=1THEN720 :rem 8
681 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
;E2$:CLOSE15:GOTO720 :rem 103
682 GOTO720 :rem 115
690 PRINT"{CLR}{RVS}*** LOAD ***
{ 2 GIU'}" :rem 212
800 INPUT"{ 2 GIU'} NOME DEL FILE";F$
: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$):SYS65469 :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870
:rem 111
865 PRINT"{GIU'}OPERAZIONE CONCLUSA.":GOT
O310 :rem 142
870 PRINT"{GIU'}ERRORE DI CARICAMENTO.RIP
ROVA.{GIU}":IFDV=1THEN800 :rem 43
880 OPEN15,8,15:INPUT#15,E1$,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
:rem 23

```

MLX nuova versione per C64

```

100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
POKE53281,1:POKE53280,1 :rem 73
101 POKE788,52:REM DISABILITA RUN/STOP
:rem 89
110 PRINT"{RVS}{ 40 SPAZI}"; :rem 176
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}{<*>]
E{RVS}{DES}{DES}{ 2 SPAZI}{<*>]{OFF}
[<*>]E{RVS}E{RVS}{ 13 SPAZI}";
:rem 250
130 PRINT"{RVS}{ 15 SPAZI}{DES}{<N>]{<H>]
{DES} { 2 DES} {OFF}E{RVS}E[<*>]{OFF}
[<*>]{RVS}{ 13 SPAZI}"; :rem 220
140 PRINT"{RVS}{ 40 SPAZI} " :rem 120
200 PRINT"{ 2 GIU'}{PUR} UN PROGRAMMA
{ 3 SPAZI}PER{ 3 SPAZI}L'INTRODUZIONE
DI{ 2 SPAZI}ROUTINE IN LINGUAGGIO";

```

```

:rem 217
205 PRINT" MACCHINA A PROVA{ 17 SPAZI}DI
{ 2 SPAZI}ERRORE{ 3 GIU'}" :rem 244
210 PRINT"[<5>]{ 2 SU} INDIRIZZO DI PARTE
NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
31+119*F) :rem 85
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
THENGOSUB3000:GOTO210 :rem 235
225 PRINT:PRINT:PRINT :rem 180
230 PRINT"[<5>]{ 2 SU} INDIRIZZO CONCLUSI
VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$(3
1+119*F) :rem 92
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
THENGOSUB3000:GOTO230 :rem 183
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
LUSIVO<INDIRIZZO INIZIALE" :rem 200

```

```

255 IFE<STHENGOSUB1000:GOTO230 :rem 119
260 PRINT:PRINT:PRINT :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0 :rem 225
310 PRINTRIGHT$("0000"+MID$(STR$(AD),2),5 :rem 234
);":":FORJ=1TO6
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320 :rem 228
390 IFN=-211THEN710 :rem 62
400 IFN=-204THEN790 :rem 64
410 IFN=-206THENPRINT:INPUT"{GIU'}_INSERI :rem 13
RE IL NUOVO INDIRIZZO";ZZ
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT" :rem 105
{RVS}ESCE DAL CAMPO DI VALORI INDICAT
O"
415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000 :rem 202
:GOTO410
417 IFN=-206THENAD=ZZ:PRINT:GOTO310 :rem 238
420 IFN<>-196THEN480 :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT" :rem 183
{ 9 SPAZI}A";:INPUTT
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO" :rem 147
;S;" MASSIMO";E;"![<5>]":GOTO430
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000 :rem 30
0"+MID$(STR$(I),2),5);":":
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$("00 :rem 66
"+MID$(STR$(N),2),3);":":
460 GETA$:IFA$>""THENPRINT:PRINT:GOTO310 :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN :rem 50
T:GOTO310
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK :rem 200
SUM=(CKSUM+A(I))AND255:NEXT
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20) :rem 234
:rem 255
515 IFN=CKSUMTHEN530 :rem 157
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN :rem 157
SERITA IN MANIERA"
525 PRINT"ERRATA. RIPETERE[<5>]":PRINT:GO :rem 27
SUB1000:GOTO310
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54 :rem 227
272,0:POKE54273,0
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT"[<X>]"; :rem 79
581 GETA$:IFA$=""THEN581 :rem 95
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44 :rem 229
ORA=32THEN670
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=LANDT=44THENN=-1:PRINT" :rem 172
{SIN} {SIN}";:GOTO690
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$;: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
691 FORI=1TO3:T=PEEK(S%-I) :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT :rem 205

```

```

700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN :rem 7
710 PRINT"{CLR}{RVS}*** SAVE *** :rem 236
{ 3 GIU'}"
720 INPUT"{GIU'}_NOME DEL FILE";F$ :rem 199
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO :rem 128
O {RVS}D{OFF}ISCO: (N/D)"
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$ :rem 3
):POKE782,ZK/256
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 109
T$):SYS65469
763 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 69
66
765 POKE254,S/256:POKE253,S-PEEK(254)*256 :rem 12
:POKE780,253
766 POKE782,E/256:POKE781,E-PEEK(782)*256 :rem 124
:SYS65496
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780 :rem 111
775 PRINT"{GIU'}FATTO.":END :rem 194
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!" :rem 104
:IFDV=1THEN720
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$ :rem 103
;E2$:CLOSE15:GOTO720
790 PRINT"{CLR}{RVS}*** LOAD *** :rem 212
{ 2 GIU'}"
800 INPUT"{ 2 GIU'}_NOME DEL FILE";F$ :rem 215
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO :rem 127
O {RVS}D{OFF}ISCO: (N/D)"
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$ :rem 2
):POKE782,ZK/256
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 107
T$):SYS65469
845 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 70
66
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870 :rem 111
865 PRINT"{GIU'}FATTO.":GOTO310 :rem 184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI! :rem 19
{GIU}":IFDV=1THEN800
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$ :rem 102
;E2$:CLOSE15:GOTO800
1000 REM CICALINO :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278, :rem 207
165
1002 POKE54276,33:POKE54273,6:POKE54272,5 :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54 :rem 202
273,0:POKE54272,0:RETURN
2000 REM CAMPANELLO :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2 :rem 152
47
2002 POKE54276,17:POKE54273,40:POKE54272, :rem 86
0
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN :rem 57
3000 PRINTC$;"{RVS} NON IN PAGINA ZERO O :rem 240
SU{DES}ROM ":GOTO1000

```

Soft Dealer

La rivista per chi vende personal computer - software - libri - accessori

Quali sono i pacchetti e i libri di informatica più venduti in Italia?
È proprio vero che tutti vogliono un home computer?
Lotus 1-2-3 è solo una moda?
Apple II C è veramente compatibile con II E?
Chi copia il software?
Come fare business con il software?

A QUESTE
E MOLTE ALTRE
DOMANDE RISPONDE **Soft Dealer**

Il primo mensile italiano per chi vende home e personal computer, software, libri e accessori di informatica. Uno strumento utilissimo per chiunque si interessi di personal e home computer. Su **Soft Dealer**, il nuovo mensile della PROGRAM EDITRICE, inviato solo in abbonamento, troverete tutte le notizie più utili, recensioni, test, prove di software, articoli di marketing, classifiche dei migliori programmi esistenti sul mercato.

UN ABBONAMENTO A **Soft Dealer**
È UN INVESTIMENTO PER IL FUTURO. APPROFITTA DELL'ECCEZIONALE OFFERTA DI LANCIO.
RICEVERAI GRATIS I PRIMI TRE NUMERI DEL 1984.

Per abbonarsi basterà compilare il modulo pubblicato qui sotto, allegando assegno di L. 25.000, e inviarlo a PROGRAM EDITRICE S.r.l. Via Melchiorre Gioia, 114 - 20125 Milano.

SOFT DEALER è un'idea

PROGRAM
s.r.l. **editrice**

Uffici Tecnici e Commerciali:
20125 Milano - Via M. Gioia, 114
Tel. (02) 603415-680851

- Desidero sottoscrivere un abbonamento a 10 numeri (L. 25.000) alla rivista **SOFT DEALER**, da gennaio a dicembre 1985.
In questo caso mi saranno inviati in omaggio i primi tre numeri del 1984.
- Allego assegno di L. 25.000 intestato a PROGRAM EDITRICE
n° Banca
- Ho effettuato pagamento tramite vaglia postale.

NOME

COGNOME

AZIENDA

VIA N°

C.A.P. CITTA'





SuperBasic

di **M.C. Kees**

trad. e adatt. di

N. Mandelli, F. Palazzi, O. Calvi

SUPERVIC è lieta di presentare SUPERBASIC, un fantastico programma interamente in linguaggio macchina che aggiunge ben 40 (diconsi quaranta!) nuovi comandi al BASIC del C64.

Con SUPERBASIC avrete finito di caricare di volta in volta in memoria il particolare tool che vi serve in una specifica occasione: i nuovi comandi che aggiunge si occupano di tutto: sprite, colori, grafica in alta risoluzione, musica, organizzazione della memoria, etc. Possono essere usati sia in modo diretto che da programma, e sono formati dal carattere "I" più un codice mnemonico di quattro lettere. L'unico prezzo da pagare è il non utilizzo del comando STOP, che può comunque essere agevolmente sostituito dal comando END. Le istruzioni aggiunte dal SUPERBASIC possono essere suddivise in sette categorie principali:

- standard modificate
- per il controllo dei suoni
- per il controllo dei colori
- per il controllo degli sprite
- pr il controllo del chip VIC II
- per l'uso del banco di memoria 3
- per il controllo della grafica hi-res

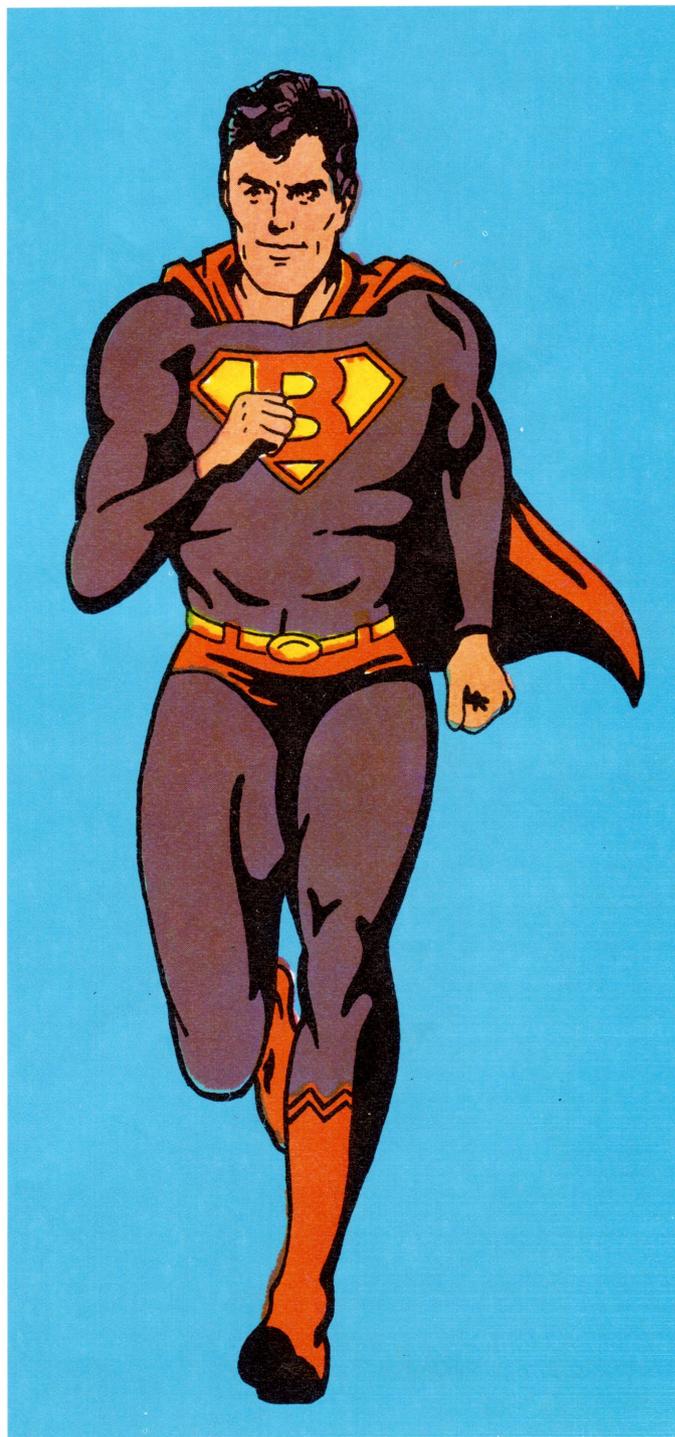
Mai più quindi lunghe serie di POKE per ottenere un suono od un lentissimo plottaggio di una curva in pagina grafica! Basta un comando e si ottiene ciò che si vuole con la velocità del linguaggio macchina.

Note per la battitura di SuperBasic

Per caricare SUPERBASIC è necessario usare MLX (vedi in altra parte della rivista). Poiché SUPERBASIC viene caricato nell'area di memoria normalmente occupata dal BASIC (previo un suo rilocamento in RAM), bisogna usare un accorgimento particolare per evitare che SUPERBASIC ricopra una parte di MLX. La procedura, da seguire attentamente, è la seguente:

- 1) Spegner e riaccendere il C64
- 2) Digitare le seguenti istruzioni e premere RETURN:
POKE44,22: POKE642,22: POKE5632,0: NEW
- 3) Caricare in memoria MLX e dare il RUN
- 4) Quando MLX chiede gli indirizzi di partenza e di fine, digitare rispettivamente 2049 e 5306
- 5) Terminato il lavoro di battitura del SUPERBASIC, usate il SAVE dell'MLX per registrarlo su nastro o su disco. Se non si termina il lavoro in una volta sola, occorre ripetere tutta la procedura descritta (punti 1-4), prima di riprendere la digitazione.

Una volta salvato l'intero SUPERBASIC su disco o nastro,



potrete richiamarlo in memoria con un normale LOAD, e poi dare il RUN, senza più bisogno delle POKE prima descritte.

Utilizzo

Quando si fa partire il Superbasic, viene copiato il Basic dalla ROM nei corrispondenti indirizzi di RAM (\$A000 - \$BFFF), vengono poi modificati alcuni comandi e vengono aggiunte le routines in linguaggio macchina per la gestione di quelli nuovi. Queste routines vengono caricate dall'indirizzo \$C000 all'indirizzo \$CC00.

Il Superbasic è quindi incompatibile con i programmi che risiedono negli stessi indirizzi, può comunque coesistere con il DOS 5.1.

All'interno dei programmi che richiedono il Superbasic non può essere utilizzato il comando STOP che può essere sostituito dal comando END.

Tutti gli operandi dei comandi Superbasic possono essere espressi sia in valore assoluto che usando variabili impostate col valore desiderato.

Listando un programma contenente comandi Superbasic con lo stesso inattivo, il carattere "[" verrà listato come STOP.

Premendo RUN/STOP-RESTORE si disabilita il Superbasic che può essere riattivato con il comando:

POKE 1,54

COMANDI

Comandi standard modificati

RESTORE: Può essere seguito da un numero di linea o da una variabile contenente un numero di linea.

Questo fa sì che una READ eseguita dopo il comando "RESTORE n" leggerà la linea di DATA di numero uguale o superiore ad "n".

LIST: il tasto SHIFT è in grado di interrompere la visualizzazione della lista di un programma sullo schermo finché è tenuto premuto.

GOTO

GOSUB

ON: Tutte le possibili forme di scrittura di questi comandi ammettono che il numero di linea sia indicato con una variabile.

Es. 10 K = 1000

20 GOSUB K

ASC(\$str): La funzione da come risultato "0" se applicata ad una stringa nulla.

Comandi per il controllo dei suoni

[SSND (Set-up SouND)

Il comando [SSND produce un suono e ne determina le caratteristiche.

Il volume del suono viene impostato al valore massimo.

[SSND voce,ad,sr,onda,freq,[,pfreq]

voce Numero della voce usata. Può avere un valore compreso tra 1 e 3

ad Valori di ATTACK e DECAY. Può assumere un valore tra 0 e 255.

sr Valori di SUSTAIN e RELEASE. Può assumere un valore tra 0 e 255.

onda Definisce la forma d'onda ovvero il timbro del suono. Possono essere definite quattro forme d'onda e per ogni forma d'onda 2 valori che fanno partire o fermare il suono.

FORMA D'ONDA	START	STOP
Triangolare	17	16
A dente di sega	33	32
Rettangolare (pulsante)	65	64
Rumore	129	128

freq Definisce la nota. Il suo valore può variare tra 0 e

65535.

Far riferimento al "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE", appendice E pag. 384 per il valori di ogni nota.

pfreq Va usato in congiunzione alla forma d'onda rettangolare (64/65). Definisce la lunghezza del ciclo di pulsazione ovvero la durata della parte positiva dell'onda. Può assumere un valore tra 0 e 12228.

NOTA SU AD/SR

Quando una nota viene emessa, passa da un volume 0 a un volume di picco in un certo tempo (controllato da ATTACK), per poi scendere ad un valore di regime (controllato da SUSTAIN) in un intervallo di tempo (controllato da DECAY) e si smorza infine a volume 0 in un certo tempo (controllato da RELEASE).

I valori di ATTACK, DECAY e RELEASE determinano una durata e variano tra 0 e 15. Vedi tabellina durate a pag. 198 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE".

SUSTAIN invece definisce il volume di mantenimento della nota rispetto al volume di picco. Varia tra 0 e 15 con scala lineare dove 0 significa "suono assente" e 15 "volume di picco".

Il valore AD è determinato dalla formula:

$$AD = ATTACK * 16 + DECAY$$

con formula analoga si determina SR

$$SR = SUSTAIN * 16 + RELEASE$$

[PLAY

Il comando [PLAY è una forma abbreviata di [SSND, va usata per modificare la nota. Assume che i valori AD/SR siano stati precedentemente impostati dalla [SSND.

[PLAY vonda, freq[, pfreq]

vonda Parametro che definisce sia la forma d'onda che la voce da usare. Fare riferimento agli operandi VOCE ed ONDA della [SSND per i valori ammessi.

Il contenuto di VCONDA è definito dalla seguente formula:

$$VCONDA = ONDA * 256 + VOCE$$

freq Definisce la nota. Il suo valore può variare tra 0 e 65535. Far riferimento al "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE", appendice E pag. 384 per i valori di ogni nota.

pfreq Va usato in congiunzione alla forma d'onda rettangolare (64/65). Definisce la lunghezza del ciclo di pulsazione ovvero la durata della parte positiva dell'onda. Può assumere un valore tra 0 e 12228.

Comandi per il controllo dei colori

[BKGD (Back Ground color)

Il comando [BKGD serve a definire il colore di fondo dello schermo.

[BKGD col

col Numero fra 0 e 15 che definisce il colore

[EXTC (EXTernal Color)

Il comando [EXTC serve a definire il colore del bordo dello schermo.

[EXTC col

col Numero fra 0 e 15 che definisce il colore

[FCOL (Fill COlor memory)

Il comando [FCOL riempie il buffer video della memoria colore con la tinta richiesta. Fa diventare dello stesso colore tutto lo schermo in high resolution o tutti i caratteri già presenti sullo schermo.

[FCOL col

col Numero fra 0 e 15 che definisce il colore

[SCRN /SCReeN colors set)

Il comando [SCRN serve a definire i colori di fondo, del bordo e dei caratteri in una sola operazione.

[SCRN bgc,boc, chc

bgc Numero fra 0 e 15 che definisce il colore di fondo dello schermo.

boc Numero fra 0 e 15 che definisce il colore del bordo dello schermo.

chc Numero fra 0 e 15 che definisce il colore dei caratteri sullo schermo.

[BKG4 (Back Ground colors)

Il comando [BKG4 definisce tutti i 4 colori di fondo dello schermo usabili in EXTENDED COLOR MODE e MULTI COLOR BIT MAP MODE.

[BKG4 col0, col1, col2, col3

col0-3 Numeri fra 0 e 15 che definiscono rispettivamente il colore di fondo dello schermo (col0), il colore background 1 (col1), il colore background 2 (col2) e il colore dei caratteri (col3).

Per maggiori informazioni vedi i paragrafi MULTICOLOR MODE GRAPHICS e EXTENDED BACKGROUND COLOR MODE da pag. 115 a 121 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE".

Comandi per il controllo degli sprites

[DSPR (Define SPRite)

Il comando [DSPR è usato per definire le caratteristiche di uno sprite.

[DSPR spr, blk, xexp, yexp, xpos, ypos, multi, col[, mc0, mc1]

spr Numero dello sprite definito, può variare tra 0 e 7

blk Sprite pointer. Può avere un valore tra 0 e 255.

Siccome per definire uno sprite occorrono 64 Kbytes (1 blocco), lo sprite pointer indica in quale blocco del banco da 16K visibile dal chip di gestione sprite VIC II risiede lo sprite stesso.

Per esempio, se il VIC II punta al banco 1 e blk = 14, lo sprite è definito alla posizione di memoria:

$$1 * 16384 + 14 * 64 = 17280$$

xexp Espansione orizzontale dello sprite.

0 = non espanso

1 = espanso

yexp Espansione verticale dello sprite.

0 = non espanso

1 = espanso

xpos Posizione iniziale dell'angolo superiore sinistro dello sprite sull'asse orizzontale.

Può assumere un valore da 0 a 512 (0 a sinistra)

Fare riferimento alla pag. 143 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE" per i valori che rendono visibile lo sprite sullo schermo.

ypos Posizione iniziale dell'angolo superiore sinistro dello sprite sull'asse verticale.

Può assumere un valore da 0 a 256 (0 in alto)

Fare riferimento alla pag. 143 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE" per i valori che rendono visibile lo sprite sullo schermo.

multi Definisce se lo sprite è multicolor oppure di un unico colore.

0 = colore singolo

1 = multicolor

Se multi = 1 devono essere definiti anche i parametri opzionali mc0 e mc1.

col Definisce il colore dello sprite o il colore base dello sprite se multicolor. Valore tra 0 e 15

mc0 Definisce il secondo colore dello sprite multicolor. Valore tra 0 e 15

mc1 Definisce il terzo colore dello sprite multicolor. Valore tra 0 e 15

Nota sul colore degli sprites

Sprites di un solo colore:

I bits in ON dello sprite assumono il colore definito dal pa-



rametro col.

I bits in OFF hanno il colore di fondo dello schermo.

Sprites multicolor:

I bits degli sprite vengono considerati a coppie. Se il valore di una coppia è "00" questa assume il colore di fondo dello schermo.

Le coppie di valore "10" assumono il colore definito dal parametro col. Le coppie di valore "01" e "11" assumono rispettivamente i colori definiti nei parametri mc0 e mc1.

[ESPR (Enable SPRite)

Il comando [ESPR serve per abilitare lo sprite ovvero a renderlo visibile.

[ESPR spr

spr Numero dello sprite che si vuole abilitare. Può variare tra 0 e 7

[KSPR (Kill SPRite)

Il comando [KSPR serve per disabilitare lo sprite ovvero a renderlo invisibile.

[KSPR spr

spr Numero dello sprite che si vuole disabilitare. Può variare tra 0 e 7

[BSPP (Background/SPrite Priority)

Il comando [BSPP serve a definire la priorità di uno sprite rispetto al "fondo", ovvero definisce se uno sprite passerà "dietro" ai caratteri dello schermo oppure "davanti".

[BSPP spr, sel

spr Numero tra 0 e 7 che definisce lo sprite.

sel Priorità dello sprite rispetto al background.

0 = lo sprite ha priorità maggiore e nasconde i caratteri

1 = lo sprite ha priorità più bassa e ne è nascosto

[MOVE (MOVE sprite)

Il comando [MOVE serve a muovere lo sprite sullo schermo.

[MOVE spr, xpos, ypos

spr Numero tra 0 e 7 che definisce lo sprite.

xpos Posizione dell'angolo superiore sinistro dello sprite sull'asse orizzontale.

Può assumere un valore da 0 a 512 (0 a sinistra)

Fare riferimento alla pag. 143 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE" per i valori che rendono visibile lo sprite sullo schermo.

ypos Posizione dell'angolo superiore sinistro dello sprite sull'asse verticale.

Può assumere un valore da 0 a 256 (0 in alto)

Fare riferimento alla pag. 143 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE" per i valori che rendono visibile lo sprite sullo schermo.

Comandi per il controllo del chip VIC II

Il chip VIC II può vedere solo 16K di memoria per volta, la

memoria è perciò divisa in 4 banchi di 16K ciascuno. Normalmente il chip accede ai primi 16K, ovvero al banco "0".

All'interno del banco selezionato, può essere definito quale blocco da 1K costituisce la memoria dello schermo (buffer video), normalmente il blocco "1".

Il chip VIC II deve inoltre selezionare il blocco da 2K che definisce il CHARACTER SET disponibile.

[BANK (define BANK)

Il comando [BANK definisce quale banco di memoria è visibile al VIC II.

[BANK sel

sel Numero che indica il blocco. Può variare tra 0 e 3

[VS1K (video Screen 1K)

Il comando [VSK1 determina quale dei 16 blocchi da 1K, disponibili all'interno di un banco, viene usato per il buffer video.

[VS1K sel

sel Numero che indica il blocco. Può variare tra 0 e 15.

[CB2K (Character Block 2K)

Il comando [CB2K seleziona quale degli 8 blocchi da 2K, disponibili all'interno di un banco, viene usato per il character set.

[CBSK sel

sel Numero che indica il blocco. Può variare tra 0 e 7.

NOTA: [CBSK è usato anche per selezionare quale dei 2 blocchi da 8K disponibili è usato per la BIT MAP SCREEN memory.

I valori di SEL da 0 a 3 selezionano il primo blocco mentre i valori da 4 a 7 selezionano il blocco superiore.

I 3 comandi [BANK, [VS1K e [CB2K vanno usati insieme e in modo coordinato, facendo attenzione nel selezionare i blocchi usati. Il sistema può "cadere" se si selezionasse (ad esempio) per la memoria video la pagina "0" (\$0000-\$0400). Anche i banchi "2" e "3" devono essere utilizzati con cautela.

Spostando la memoria video, si può abbassare l'entry point dei programmi Basic da \$0800 e \$0400 guadagnando 1K.

Comandi per l'uso del BANCO 3

Quando viene selezionato il banco 3, il VIC II vede la RAM dall'indirizzo \$C000 all'indirizzo \$FFFF, ignorando la ROM di pari indirizzo che è però indispensabile al Basic.

Il Superbasic permette l'uso del banco 3 per allocarvi la memoria video ("[VS1K 3" dall'indirizzo \$CC00 che è quello successivo al massimo indirizzo occupato dal Superbasic). La RAM da \$D000 può essere usata per character set, definizione sprites, bit map screen memory, etc. Per usare questi indirizzi bisogna abilitare, usare e disabilitare la RAM.

[HRAM (High RAM usage)

Il comando [HRAM permette che altri comandi Superbasic possano essere eseguiti usando il banco 3 di memoria.

[HRAM supcom [param]

supcom È un qualsiasi comando Superbasic ad eccezione del comando [MXGR.

param È la lista dei parametri richiesti dal comando "supcom".

[STUF (STore in core)

Il comando [STUF è un equivalente della POKE e funziona esattamente allo stesso modo. A differenza della POKE, può essere eseguito nel banco 3 come "subcom" del comando [HRAM.

[STUF loc,val

[HRAMLOOK loc, val

loc È la posizione di memoria dove viene messo il dato "val".

val È il dato da mettere nella posizione di memoria "loc".

[LOOK (LOOK in core)

Il comando [LOOK è un equivalente della PEEK; ha però una sintassi diversa perché richiede l'uso di una variabile dove mettere il valore estratto dalla memoria. Può essere eseguito nel banco 3 come "subcom" del comando [HRAM.

[LOOK loc,val

[HRAM100K loc, val

loc È la posizione di memoria da dove viene prelevato il dato "val".

val È la variabile nella quale viene messo il dato prelevato dalla memoria "loc".

Comandi per il controllo della grafica

[ECGR (Extended Color mode GRaphic)

Il comando [ECGR seleziona la gestione della grafica in extended color mode.

[ECGR switch

switch Abilitazione/disabilitazione extended color mode

0 = disabilitato

1 = abilitato

[MCGR (Multi Color mode GRaphic)

Il comando [MCGR seleziona la gestione della grafica in multi color mode.

[MCGR switch

switch Abilitazione/disabilitazione multi color mode

0 = disabilitato

1 = abilitato

[BMGR (Bit Map mode GRaphic)

Il comando [BMGR seleziona la gestione della grafica in bit map mode.

[BMGR switch

switch Abilitazione/disabilitazione bit map mode

0 = disabilitato

1 = abilitato

NOTA: [BMGR e [MCGR possono essere usati insieme per ottenere un bit map graphics in multi color mode.

[ECGR e [BMGR, se usati insieme, mettono a blank lo schermo velocizzando l'esecuzione dei programmi.

[DLCS (DownLoad Character Set)

Il comando [DLCS serve a copiare in RAM il character set. Possono essere copiati i caratteri minuscoli, maiuscoli o entrambi.

[DLCS set,addr

set Indica il character set da copiare:

0 = set maiuscolo

1 = set minuscolo

2 = entrambi i sets

addr È l'indirizzo di partenza del nuovo character set.

Deve essere allineato ai 2K, a meno che non si desideri cambiare l'ordine di definizione dei caratteri all'interno del set.

Usando addr = 53248 il set sarà copiato nella RAM sottostante il character set standard in ROM nel BANCO 3.

Potrà essere modificato con l'uso della [HRAMSTUF.

[FBMS (Fill Bit Map Screen)

Il comando [FBMS va usato in BIT MAP MODE. Riempie il buffer video high resolution con la configurazione di bits prescelta.

[FBMS byte

byte È una variabile contenente un valore numerico tra 0 e 255 o il numero stesso.

[FSCR (Fill SCReen)

Il comando [FSCR riempie il buffer video con il carattere specificato. Può essere usato anche in BIT MAPE MODE dove la memoria video serve a determinare il colore dello schermo high resolution.

[FSCR byte

byte È una variabile contenente la rappresentazione ASCII del carattere da mettere sul video o che definisce il colore da usare, oppure il numero stesso.

[PLOT (PLOT pixel)

Il comando [PLOT va usato in BIT MAPE MODE, rende visibile uno dei 320 * 200 punti dello schermo.

[PLOT xpos,ypos

xpos Posizione del punto sull'asse orizzontale. Può avere un valore tra 0 e 319.

ypos Posizione del punto sull'asse verticale. Può avere un valore tra 0 e 199, dove "0" è il primo "raster" (riga scandita dal pennello elettronico) in alto.

[CLPX (CLear PiXel)

Il comando [CLPX va usato in BIT MAPE MODE, cancella uno dei 320 * 200 punti dello schermo.

[CLPX xpos,ypos

xpos Posizione del punto sull'asse orizzontale. Può avere un valore tra 0 e 319.

ypos Posizione del punto sull'asse verticale. Può avere un valore tra 0 e 199, dove "0" è il primo "raster" (riga scandita dal pennello elettronico) in alto.

[FLIP (FLIP flop pixel)

Il comando [FLIP va usato in BIT MAP MODE, cambia lo stato di uno dei 320 * 200 punti dello schermo ovvero se è visibile lo rende invisibile e viceversa.

[FLIP xpos,ypos

xpos Posizione del punto sull'asse orizzontale. Può avere un valore tra 0 e 319.

ypos Posizione del punto sull'asse verticale. Può avere un valore tra 0 e 199, dove "0" è il primo "raster" (riga scandita dal pennello elettronico) in alto.

[MCPL (Multi Color PLoT pixel)

Il comando [MCPL va usato in multicolor BIT MAP MODE, e definisce il colore e la posizione di uno dei 160 * 200 punti dello schermo.

[MCPL xpos,ypos,sel

xpos Posizione del punto sull'asse orizzontale. Può avere un valore tra 0 e 159.

ypos Posizione del punto sull'asse verticale. Può avere un valore tra 0 e 199, dove "0" è il primo "raster" (riga scandita dal pennello elettronico) in alto.

sel Può avere un valore tra 0 e 3 con i seguenti significati:

- 0 = colore del background screen
- 1 = colore definito nel semibyte di destra del buffer video
- 2 = colore definito nel semibyte di sinistra del buffer video
- 3 = colore definito nel buffer colore video.

Esempio: Il buffer video è stato riempito con [FSCR 16

ovvero - semibyte di sinistra = 1 = bianco
- semibyte di destra = 0 = nero

Il buffer colore video è stato riempito di colore blu con [FCOL6

Il colore di sfondo dello schermo è stato impostato a



giallo con

[BKGD7

Il comando

- [MCPL 30,40,0 genera un pixel giallo (invisibile)
- [MCPL 40,50,1 genera un pixel nero 0000
- [MCPL 50,60,2 genera un pixel bianco 0001
- [MCPL 70,80,3 genera un pixel blu

[DRAW (DRAW linea)

Il comando [DRAW è usato in BIT MAP MODE (anche multicolor) per disegnare o cancellare un segmento sullo schermo.

[DRAW type,xpos1,ypos1,xpos2,ypos2

type Opzione di scrittura o di cancellazione

0 = cancellazione

1 = scrittura

xpos1 Posizione del punto di inizio del segmento sull'asse orizzontale. Può avere un valore tra 0 e 319.

ypos1 Posizione del punto di inizio del segmento sull'asse verticale. Può avere un valore tra 0 e 199.

xpos2 Posizione del punto di fine del segmento sull'asse orizzontale. Può avere un valore tra 0 e 319.

ypos2 Posizione del punto di fine del segmento sull'asse verticale. Può avere un valore tra 0 e 199.

[HRCS (High Resolution Character Set)

Il comando [HRCS memorizza l'indirizzo di inizio del character set da usare in BIT MAP MODE.

Il character set può anche non essere allineato ai 2K e può essere il character set standard, in quest'ultimo caso l'indirizzo deve essere 53248 che seleziona il character set MAIUSCOLO in ROM.

[HRCS addr

addr Indirizzo di inizio del character set

[CODE (CODE character in screen code)

Il comando [CODE permette di tradurre una stringa dalla codifica ASCII a quella SCREEN DISPLAY CODE, che è la codifica usata dai comandi [CHAR e [CHR per scrivere dei caratteri sullo schermo in bit map mode.

[CODE str\$

str\$ È il nome di una variabile che contiene la stringa da convertire. Dopo l'esecuzione il contenuto della stringa è in SCREEN CODE e non più in ASCII; va quindi salvata prima se si vuole riutilizzarla.

I caratteri di controllo RVS-ON e RVS-OFF possono essere usati nella stringa per selezionare il set maiuscolo (OFF) o il set minuscolo (ON).

Ogni altro carattere di controllo non deve essere usato nella stringa perché può dar luogo a risultati imprevedibili.

[CHAR (write CHARacter in bit map mode)

Il comando [CHAR permette di scrivere un carattere sullo schermo in bit map mode.

[CHAR byte,xpos,ypos

byte È la rappresentazione in screen code del carattere da mettere sullo schermo, come specificato nell'appendice B di pag. 376 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE". La variabile byte può essere ottenuta

facendo, una dopo l'altra, le necessarie [CODE delle MID\$ sulla stringa origine.

xpos È la posizione sull'asse orizzontale dell'angolo superiore sinistro della matrice carattere.

ypos È la posizione sull'asse verticale dell'angolo superiore sinistro della matrice carattere.

[CHRX (write CHaRacter in bit map mode with eXclusive OR)

Il comando [CHRX permette di scrivere un carattere sullo schermo in bit map mode facendo exclusive OR con quanto già esistente sullo schermo in quella posizione. Può perciò essere utilizzato per cancellare un carattere già presente sullo schermo.

[CHRX byte,xpos,ypos

byte È la rappresentazione in screen code del carattere da mettere sullo schermo, come specificato nell'appendice B di pag. 376 del "COMMODORE 64 PROGRAMMER'S REFERENCE GUIDE". La variabile byte può essere ottenuta facendo, una dopo l'altra, le necessarie [CODE delle MID\$ sulla stringa origine.

xpos È la posizione sull'asse orizzontale dell'angolo superiore sinistro della matrice carattere.

ypos È la posizione sull'asse verticale dell'angolo superiore della matrice carattere.

[MXGR (MiXed mode Graphic set-up)

Il comando [MXGR crea una interrupt routine che consente di gestire contemporaneamente lo schermo in 2 modi grafici diversi.

Ogni volta che il "raster register" (il registro che conta le righe orizzontali pennellate sullo schermo) eguagli uno dei valori indicati, viene chiamata la interrupt routine che sostituisce il valore indicato in un registro del chip VIC II.

Ricordare che la parte visibile dello schermo va dal raster 51 al raster 251.

[MXGR reg,mask,rast1,val1,rast2,val2

reg È uno dei registri del VIC II. Il registro è indicato dall'indirizzo all'interno del chip. Siccome l'indirizzo iniziale del chip è 53248 e ad esempio l'indirizzo del registro che controlla il colore di fondo dello schermo è 53281, l'operando reg in questo caso è "33" (53281—53248).

mask Controlla quali bit del registro vengono modificati. Può avere un valore tra 0 e 255.

I bit del registro che vengono modificati sono quelli corrispondenti ai bit in OFF della maschera (mask).

Per esempio la maschera "240" ((\$F0) — (11110000)) permette la modifica del solo semibyte di destra del registro.

rast1 Primo valore di raster per il quale viene attivata l'interrupt routine.

val1 Valore che viene inserito nei bit del registro (reg) corrispondenti ai bit in off della maschera (mask) quando il raster register eguaglia il valore specificato in rast1.

rast2 Secondo valore di raster per il quale viene attivata l'interrupt routine.

val2 Valore che viene inserito nei bit del registro (reg) corrispondenti ai bit in off della maschera (mask) quando il raster register eguaglia il valore specificato in rast2.

Esempio: L'istruzione

[MXGR 33,240,152,10,252,0

farà sì che il fondo dello schermo (reg = 33) risulti nero nella metà superiore (raster 151) e rosa nella metà inferiore (raster 152 al raster 251).

Nota: La temporizzazione dell'interrupt routine può essere imprecisa perché la interrupt routine standard (IRQ) è sempre attiva. Ciò può causare delle ondulazione sulla linea di separazione fra i "modi" dello schermo. Per mantenere questa linea precisa occorre disabilitare la IRQ perdendo così il controllo sulla keyboard e limitando l'I/O al solo joystick.

[CMXV (Change MiXed graphic mode Values)

Il comando [CMXV permette di variare dinamicamente i valori da inserire nel registro mentre la interrupt routine creata dalla [MXGR è attiva.

[CMXV val1,val2

val1 Valore che viene inserito nei bit del registro (reg) corrispondenti ai bit in off della maschera (mask) quando il raster register eguaglia il valore specificato in rast1.

val2 Valore che viene inserito nei bit del registro (reg) corrispondenti ai bit in off della maschera (mask) quando il raster register eguaglia il valore specificato in rast2.

[KMXG (Kill MiXed Graphic mode)

Il comando [KMXG disabilita l'interrupt routine creata dalla [MXGR lasciando il registro in uno stato indeterminato. Per forzare il valore voluto nel registro usare prima la [CMXV. Non ha operandi.

[KMXG

I due comandi che seguono utilizzano la possibilità dell'integrato VIC II di eseguire lo scrolling lento dello schermo video sia in orizzontale che in verticale.

Questo tipo di movimento viene effettuato spostando l'intero schermo lateralmente o verticalmente di un solo punto per volta. Lo scrolling lento ha però come requisito la riduzione dello schermo nella direzione del movimento.

Per fare uno scrolling verticale lo schermo visibile deve essere portato da 25 a 24 righe, e per uno scrolling orizzontale da 40 a 38 caratteri per riga.

[SIZE (Screen SIZE)

Il comando [SIZE permette di selezionare il numero di righe e/o colonne che si vogliono visibili sullo schermo.

[SIZE colsel,rowsel

colsel Seleziona il numero di colonne visibili sullo schermo.

0 = 38 colonne

1 = 40 colonne

rowsel Seleziona il numero di righe visibili sullo schermo.

0 = 24 righe

1 = 25 righe

[XYSC (X-Y Scroll)

Il comando [XYSC muove l'intero schermo orizzontalmente e/o verticalmente da 1 a 7 pixel per volta.

Normalmente viene utilizzato lo spostamento di 1 pixel alla volta per realizzare lo scroll lento.

[XYSC xval,yval

xval Può assumere un valore tra 0 e 7.

Variando ad esempio il valore tra 0 e 7 o tra 7 e 0 lo schermo può essere mosso di un pixel alla volta rispettivamente verso l'alto o verso il basso.

yval Può assumere un valore tra 0 e 7.

Variando ad esempio il valore tra 0 e 7 o tra 7 e 0 lo schermo può essere mosso di un pixel alla volta rispettivamente verso sinistra o verso destra.

Esiste inoltre il comando [CATA che lista sullo schermo tutti i nuovi comandi del SUPERBASIC.

[CATA (CATAlog of command)

[CATA

Non ha operandi.

SUPERBASIC per C64

2049 : 011,008,000,000,158,050,228
2055 : 048,056,048,000,000,000,159
2061 : 000,000,000,000,000,000,013
2067 : 000,000,000,000,000,000,019

2073 :000,000,000,000,000,000,025
 2079 :000,169,039,133,001,169,030
 2085 :000,133,020,133,078,169,058
 2091 :009,133,021,169,192,133,188
 2097 :079,162,012,160,000,177,127
 2103 :020,145,078,200,208,249,187
 2109 :230,021,230,079,202,208,007
 2115 :242,160,008,169,104,032,014
 2121 :030,171,169,013,141,119,204
 2127 :002,141,120,002,169,002,003
 2133 :133,198,169,033,141,001,248
 2139 :008,169,020,141,002,008,183
 2145 :076,120,008,000,000,000,045
 2151 :000,031,147,017,017,048,107
 2157 :017,157,082,085,078,019,035
 2163 :000,000,000,000,000,169,028
 2169 :000,133,020,169,160,133,224
 2175 :021,162,032,160,000,177,167
 2181 :020,145,020,136,208,249,143
 2187 :230,021,202,208,244,162,182
 2193 :000,160,003,185,224,160,109
 2199 :157,224,160,232,200,224,068
 2205 :190,208,244,169,003,141,088
 2211 :161,168,169,192,141,162,132
 2217 :168,169,074,141,210,166,073
 2223 :169,193,141,211,166,141,172
 2229 :037,160,169,084,141,036,040
 2235 :160,169,219,141,223,160,235
 2241 :169,255,141,044,160,169,107
 2247 :194,141,045,160,169,038,178
 2253 :133,001,169,005,141,143,029
 2259 :183,169,076,141,043,169,224
 2265 :141,087,169,169,193,141,093
 2271 :045,169,141,089,169,169,237
 2277 :200,141,088,169,169,227,199
 2283 :141,044,169,096,000,000,173
 2289 :000,000,000,000,000,000,241
 2295 :000,000,000,000,000,000,247
 2301 :000,000,000,032,115,000,144
 2307 :032,158,173,032,247,183,060
 2313 :096,032,139,192,032,000,244
 2319 :192,165,020,166,002,157,205



2325 :248,007,032,000,192,165,153
 2331 :020,162,029,032,162,192,112
 2337 :032,000,192,165,020,162,092
 2343 :023,032,162,192,032,097,065
 2349 :192,032,000,192,165,020,134
 2355 :072,162,028,032,162,192,187
 2361 :032,000,192,165,020,166,120
 2367 :002,157,039,208,104,240,045
 2373 :117,032,000,192,165,020,083
 2379 :141,037,208,032,000,192,173
 2385 :165,020,141,038,208,169,054
 2391 :001,162,021,032,162,192,145
 2397 :096,032,139,192,032,000,072
 2403 :192,165,021,072,165,020,222
 2409 :072,032,000,192,165,002,056
 2415 :010,170,232,165,020,157,097
 2421 :000,208,202,104,157,000,020
 2427 :208,104,162,016,032,162,039
 2433 :192,169,000,141,030,208,101
 2439 :141,031,208,096,032,000,131
 2445 :192,165,020,041,007,133,187
 2451 :002,170,169,001,224,000,201
 2457 :240,004,010,202,208,252,045
 2463 :133,078,096,164,078,201,141
 2469 :000,240,006,152,029,000,080
 2475 :208,208,006,152,073,255,049
 2481 :061,000,208,157,000,208,043
 2487 :096,000,007,014,032,019,095
 2493 :199,240,150,032,000,192,234
 2499 :165,020,041,003,170,189,015
 2505 :183,192,133,078,169,212,144
 2511 :133,079,032,000,192,165,040
 2517 :020,160,005,145,078,032,141
 2523 :000,192,165,020,160,006,250
 2529 :145,078,032,028,193,165,098
 2535 :020,133,002,160,004,145,183
 2541 :078,032,037,193,169,015,249
 2547 :141,024,212,096,032,000,236
 2553 :192,165,020,041,003,170,072
 2559 :189,183,192,133,078,169,175
 2565 :212,133,079,165,021,133,236
 2571 :002,169,000,160,004,145,235
 2577 :078,032,037,193,165,002,012
 2583 :160,004,145,078,096,169,163
 2589 :000,160,004,145,078,076,236
 2595 :000,192,032,000,192,165,104
 2601 :021,160,001,145,078,165,099
 2607 :020,136,145,078,165,002,081
 2613 :201,065,208,016,032,000,063



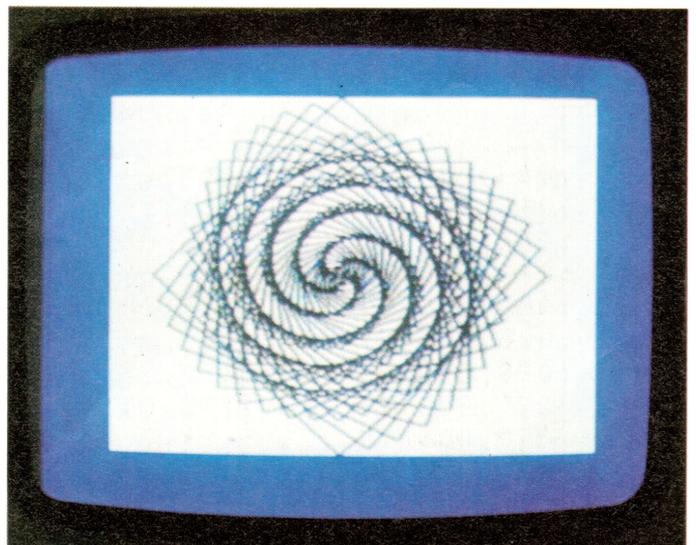
2619 :192,165,021,041,015,160,141
2625 :003,145,078,165,020,136,100
2631 :145,078,096,173,141,002,194
2637 :208,251,076,044,168,076,132
2643 :029,168,240,251,032,003,038
2649 :192,032,019,166,056,165,207
2655 :095,233,001,164,096,176,092
2661 :001,136,133,065,132,066,122
2667 :096,032,000,192,160,000,075
2673 :177,020,133,002,032,115,080
2679 :000,032,040,175,164,002,020
2685 :169,000,032,145,179,166,048
2691 :071,164,072,032,215,187,104
2697 :096,032,000,192,165,020,130
2703 :133,078,165,021,133,079,240
2709 :032,000,192,165,020,160,206
2715 :000,145,078,096,173,014,149
2721 :220,041,254,141,014,220,027
2727 :165,001,041,253,133,001,249
2733 :169,193,072,169,184,072,008
2739 :032,115,000,076,000,195,085
2745 :165,001,009,002,133,001,240
2751 :173,014,220,009,001,141,237
2757 :014,220,096,165,101,133,158
2763 :254,104,133,002,198,254,124
2769 :208,005,165,002,076,239,136
2775 :167,032,000,192,032,121,247
2781 :000,201,044,240,237,096,015
2787 :032,121,000,201,137,208,158
2793 :003,076,055,169,201,141,110
2799 :240,249,076,050,169,000,255
2805 :000,000,000,000,000,000,245
2811 :000,000,000,000,000,077,072
2817 :079,086,069,093,192,068,076
2823 :083,080,082,101,196,083,120
2829 :083,078,068,191,192,080,193
2835 :076,065,089,246,192,066,241
2841 :075,071,068,056,195,069,047
2847 :088,084,067,065,195,075,093
2853 :083,080,082,074,195,069,108
2859 :083,080,082,084,195,066,121
2865 :083,080,080,092,195,083,150
2871 :084,085,070,137,193,069,181
2877 :067,071,082,125,195,077,166
2883 :067,071,082,150,195,066,186
2889 :077,071,082,175,195,083,244
2895 :073,090,069,187,195,088,013
2901 :089,083,067,211,195,067,029
2907 :065,084,065,250,195,066,048
2913 :065,078,075,053,196,086,138
2919 :083,049,075,113,196,067,174
2925 :066,050,075,151,196,068,203
2931 :076,067,083,172,196,077,018
2937 :088,071,082,065,197,075,187
2943 :077,088,071,180,197,067,039
2949 :077,088,086,200,197,070,083
2955 :067,079,076,217,197,080,087
2961 :076,079,084,130,198,070,014
2967 :076,073,080,122,198,067,255

2973 :076,080,088,138,198,077,046
2979 :067,080,076,148,198,070,034
2985 :083,067,082,197,198,070,098
2991 :066,077,083,232,198,068,131
2997 :082,065,087,097,199,072,015
3003 :082,067,083,060,201,067,235
3009 :072,065,082,134,202,067,047
3015 :072,082,088,142,202,067,084
3021 :079,068,069,150,202,076,081
3027 :079,079,075,107,193,066,042
3033 :075,071,052,105,195,072,019
3039 :082,065,077,158,193,083,113
3045 :067,082,078,176,203,255,066
3051 :255,255,255,255,255,255,229
3057 :255,255,255,255,255,255,235
3063 :255,255,255,255,255,255,241
3069 :255,197,200,162,000,134,177
3075 :002,160,000,177,122,221,173
3081 :000,194,208,026,232,200,101
3087 :192,004,208,243,189,001,084
3093 :194,072,189,000,194,072,230
3099 :165,122,024,105,003,133,067
3105 :122,144,002,230,123,096,238
3111 :165,002,024,105,006,133,218
3117 :002,170,189,000,194,201,033
3123 :255,208,206,076,008,175,211
3129 :032,000,192,165,020,141,095
3135 :033,208,096,032,000,192,112
3141 :165,020,141,032,208,096,219
3147 :032,139,192,169,000,162,001
3153 :021,076,162,192,032,139,191
3159 :192,162,021,076,162,192,124
3165 :032,139,192,032,000,192,168
3171 :165,020,162,027,076,162,199
3177 :192,162,000,134,002,032,115
3183 :000,192,165,020,166,002,144
3189 :157,033,208,232,224,004,207
3195 :208,239,096,032,000,192,122
3201 :165,020,162,017,160,064,205
3207 :032,164,192,165,020,240,180
3213 :239,169,000,162,022,160,125
3219 :016,076,164,192,032,000,115
3225 :192,165,020,162,022,160,106
3231 :016,032,164,192,165,020,236
3237 :240,214,169,000,162,017,199
3243 :160,064,076,164,192,032,091
3249 :000,192,165,020,162,017,221
3255 :160,032,076,164,192,032,071
3261 :000,192,165,020,162,022,238
3267 :160,008,032,164,192,032,015
3273 :000,192,165,020,162,017,245
3279 :160,008,076,164,192,032,071
3285 :000,192,165,020,041,007,126
3291 :133,020,173,022,208,041,048
3297 :248,005,020,141,022,208,101
3303 :032,000,192,165,020,041,169
3309 :007,133,020,173,017,208,027
3315 :041,248,005,020,141,017,203
3321 :208,096,169,032,141,000,127

3327 :002,162,000,142,005,002,056
3333 :134,002,173,141,002,208,153
3339 :251,160,000,189,000,194,037
3345 :153,001,002,232,200,192,029
3351 :004,208,244,169,000,160,040
3357 :002,032,030,171,165,002,175
3363 :024,105,006,133,002,170,219
3369 :189,000,194,201,255,208,064
3375 :215,032,115,000,208,251,100
3381 :096,173,002,221,009,003,045
3387 :141,002,221,032,000,192,135
3393 :165,020,041,003,072,073,183
3399 :003,133,020,173,000,221,109
3405 :041,252,005,020,141,000,024
3411 :221,104,024,106,106,106,238
3417 :133,020,173,136,002,041,082
3423 :063,005,020,141,136,002,206
3429 :096,173,136,002,024,105,125
3435 :003,141,022,192,076,010,039
3441 :192,032,000,192,165,020,202
3447 :041,063,010,010,133,020,140
3453 :173,136,002,041,192,005,162
3459 :020,141,136,002,165,020,103
3465 :010,010,133,020,173,024,251
3471 :208,041,015,005,020,141,061
3477 :024,208,096,173,024,208,114
3483 :041,241,133,002,032,000,092
3489 :192,165,020,041,007,010,084
3495 :005,002,141,024,208,096,131
3501 :173,014,220,041,254,141,248
3507 :014,220,165,001,041,251,103
3513 :133,001,032,000,192,165,196
3519 :020,041,003,162,008,201,114
3525 :002,208,002,162,016,160,235
3531 :208,201,001,208,002,160,215
3537 :216,132,079,160,000,132,160
3543 :078,134,002,032,000,192,141
3549 :166,002,160,000,177,078,036
3555 :145,020,200,208,249,230,255
3561 :021,230,079,202,208,242,191
3567 :165,001,009,004,133,001,040
3573 :173,014,220,009,001,141,035
3579 :014,220,096,000,000,154,223
3585 :000,006,252,000,006,000,009
3591 :033,240,044,025,208,048,093
3597 :003,076,049,234,173,006,042
3603 :197,073,003,141,006,197,124
3609 :170,172,007,197,185,000,244
3615 :208,045,008,197,029,002,008
3621 :197,153,000,208,173,017,017
3627 :208,041,127,029,001,197,134
3633 :141,017,208,189,000,197,033
3639 :141,018,208,169,001,141,221
3645 :025,208,076,129,234,120,085
3651 :169,240,141,026,208,032,115
3657 :000,192,165,020,141,007,086
3663 :197,032,000,192,165,020,173
3669 :141,008,197,032,000,192,143
3675 :165,020,141,003,197,165,014



3681 :021,041,001,240,002,169,059
3687 :128,141,004,197,032,000,093
3693 :192,165,020,141,002,197,058
3699 :032,000,192,165,020,141,153
3705 :000,197,165,021,041,001,034
3711 :240,002,169,128,141,001,040
3717 :197,032,000,192,165,020,227
3723 :141,005,197,173,017,208,112
3729 :041,127,013,004,197,141,156
3735 :017,208,173,003,197,141,122
3741 :018,208,169,003,141,006,190
3747 :197,169,241,141,026,208,121
3753 :169,009,141,020,003,169,168
3759 :197,141,021,003,088,096,209
3765 :120,169,049,141,020,003,171
3771 :169,234,141,021,003,169,156
3777 :240,141,026,208,088,076,204
3783 :115,000,032,000,192,165,191
3789 :020,141,002,197,032,000,085
3795 :192,165,020,141,005,197,163
3801 :096,032,000,192,165,020,210
3807 :041,015,162,000,157,000,086
3813 :216,157,000,217,157,000,208
3819 :218,157,000,219,232,208,245
3825 :241,141,134,002,096,032,119
3831 :000,192,165,020,041,007,160
3837 :133,002,165,020,041,248,094
3843 :133,251,165,021,133,252,190
3849 :032,000,192,165,020,133,039
3855 :078,041,248,133,020,133,156
3861 :253,169,000,133,254,024,086
3867 :006,253,038,254,006,253,069



3873 :038,254,165,020,101,253,096
3879 :133,253,144,002,230,254,031
3885 :024,006,253,038,254,006,114
3891 :253,038,254,006,253,038,125
3897 :254,165,078,041,007,005,095
3903 :253,133,253,024,165,251,118
3909 :101,253,133,251,165,252,200
3915 :101,254,133,252,173,136,100
3921 :002,041,192,005,252,133,194
3927 :252,173,024,208,041,008,025
3933 :010,010,005,252,133,252,243
3939 :166,002,189,107,198,160,153
3945 :000,096,128,064,032,016,185
3951 :008,004,002,001,192,048,110
3957 :012,003,000,085,170,255,130
3963 :032,246,197,081,251,145,051
3969 :251,096,032,246,197,017,200
3975 :251,145,251,096,032,246,132
3981 :197,073,255,049,251,145,087
3987 :251,096,032,000,192,165,115
3993 :020,041,003,024,105,008,098
3999 :133,002,006,020,038,021,123
4005 :032,255,197,133,002,032,048
4011 :000,192,165,020,041,003,080
4017 :170,189,119,198,037,002,124
4023 :133,020,165,002,073,255,063
4029 :160,000,049,251,005,020,162
4035 :145,251,096,032,000,192,143
4041 :173,136,002,133,252,169,042
4047 :000,133,251,168,162,003,156
4053 :165,020,145,251,200,208,178
4059 :251,230,252,202,208,246,072
4065 :145,251,200,192,232,208,173
4071 :249,096,032,000,192,173,205
4077 :136,002,041,192,133,252,225
4083 :173,024,208,041,008,010,195
4089 :010,005,252,133,252,169,046
4095 :000,133,251,162,032,160,225
4101 :000,165,020,145,251,200,018
4107 :208,251,230,252,202,208,082
4113 :246,096,032,121,000,208,208
4119 :001,096,104,104,076,070,218
4125 :192,169,000,141,176,002,197
4131 :141,178,002,141,179,002,166
4137 :173,167,002,013,168,002,054
4143 :208,002,056,096,162,024,083
4149 :046,176,002,046,177,002,246
4155 :046,178,002,046,179,002,000
4161 :056,173,178,002,237,167,110
4167 :002,168,173,179,002,237,064
4173 :168,002,144,006,140,178,203
4179 :002,141,179,002,202,208,049
4185 :219,046,176,002,046,177,243
4191 :002,024,096,076,144,203,128
4197 :165,020,141,193,002,165,019
4203 :021,141,194,002,032,000,241
4209 :192,165,020,141,197,002,062
4215 :032,000,192,165,020,141,157
4221 :195,002,165,021,141,196,077

4227 :002,032,000,192,165,020,030
4233 :141,198,002,169,000,141,020
4239 :202,002,056,173,198,002,008
4245 :237,197,002,141,199,002,159
4251 :176,014,169,255,141,202,088
4257 :002,077,199,002,141,199,013
4263 :002,238,199,002,169,000,009
4269 :141,203,002,056,173,195,175
4275 :002,237,193,002,141,200,186
4281 :002,173,196,002,237,194,221
4287 :002,141,201,002,176,027,228
4293 :169,255,141,203,002,077,020
4299 :200,002,141,200,002,169,149
4305 :255,077,201,002,141,201,062
4311 :002,238,200,002,208,003,100
4317 :238,201,002,169,000,141,204
4323 :204,002,173,199,002,205,244
4329 :200,002,169,000,237,201,018
4335 :002,176,076,173,199,002,099
4341 :208,005,141,205,002,240,022
4347 :105,141,177,002,173,200,025
4353 :002,141,167,002,173,201,175
4359 :002,141,168,002,169,255,232
4365 :141,205,002,032,030,199,110
4371 :144,003,076,058,201,173,162
4377 :176,002,013,177,002,208,091
4383 :020,169,255,141,176,002,026
4389 :141,177,002,169,000,141,155
4395 :208,002,169,025,141,209,029
4401 :002,208,049,169,000,141,106
4407 :208,002,141,209,002,240,089
4413 :039,169,255,141,204,002,103
4419 :173,200,002,024,109,201,008
4425 :002,240,171,173,199,002,092
4431 :141,167,002,169,000,141,187
4437 :168,002,173,200,002,141,003
4443 :177,002,169,255,141,205,016
4449 :002,076,016,200,238,200,061
4455 :002,238,199,002,173,193,142
4461 :002,041,007,133,002,173,211
4467 :193,002,041,248,133,251,215
4473 :173,194,002,133,252,173,024
4479 :197,002,076,159,203,017,013
4485 :251,145,251,173,204,002,135
4491 :208,095,173,203,002,240,036
4497 :016,056,173,193,002,233,050
4503 :001,141,193,002,176,013,165
4509 :206,194,002,144,008,238,181
4515 :193,002,208,003,238,194,233
4521 :002,056,173,200,002,233,067
4527 :001,141,200,002,176,003,186
4533 :206,201,002,024,173,200,219
4539 :002,109,201,002,240,120,093
4545 :173,205,002,240,165,024,234
4551 :173,176,002,109,208,002,101
4557 :141,208,002,173,177,002,140
4563 :109,209,002,141,209,002,115
4569 :144,144,173,202,002,240,098
4575 :006,206,197,002,076,107,049



Super Basic

4581 :200,238,197,002,076,107,025
4587 :200,173,202,002,240,006,034
4593 :206,197,002,076,250,200,148
4599 :238,197,002,206,199,002,067
4605 :240,058,173,205,002,240,147
4611 :040,024,173,176,002,109,015
4617 :208,002,141,208,002,173,231
4623 :177,002,109,209,002,141,143
4629 :209,002,144,019,173,203,003
4635 :002,240,017,056,173,193,196
4641 :002,233,001,141,193,002,093
4647 :176,003,206,194,002,076,184
4653 :107,200,238,193,002,208,225
4659 :248,238,194,002,208,243,160
4665 :096,198,122,096,032,000,089
4671 :192,165,020,141,075,201,089
4677 :165,021,141,076,201,096,001
4683 :143,183,000,169,000,141,199
4689 :193,002,141,196,002,032,135
4695 :000,192,165,020,141,197,034
4701 :002,032,000,192,169,056,032
4707 :197,020,169,001,229,021,224
4713 :176,005,169,255,141,193,020
4719 :002,165,020,041,007,133,223
4725 :002,165,020,041,248,133,214
4731 :251,165,021,133,252,032,209
4737 :000,192,169,192,197,020,131
4743 :176,005,169,255,141,196,053
4749 :002,165,020,041,007,141,005
4755 :194,002,141,195,002,165,078
4761 :020,032,014,198,165,251,065
4767 :041,248,133,251,173,197,178
4773 :002,133,020,169,000,133,110
4779 :021,006,020,038,021,006,027
4785 :020,038,021,006,020,038,064
4791 :021,024,173,075,201,101,010
4797 :020,133,020,165,021,109,145
4803 :076,201,133,021,024,165,047
4809 :251,105,008,141,177,002,117
4815 :165,252,105,000,141,178,024
4821 :002,165,021,041,208,201,083
4827 :208,208,007,120,165,001,160
4833 :041,251,133,001,169,000,052
4839 :141,176,002,166,002,240,190
4845 :005,056,106,202,208,251,041
4851 :141,179,002,172,176,002,147
4857 :177,020,166,002,240,004,090
4863 :074,202,208,252,032,077,076
4869 :202,208,238,044,193,002,124
4875 :048,056,056,169,008,229,065
4881 :002,133,002,201,008,240,091
4887 :045,173,177,002,133,251,036
4893 :173,178,002,133,252,169,168
4899 :000,141,176,002,173,194,209
4905 :002,141,195,002,173,179,221
4911 :002,073,255,141,179,002,187
4917 :172,176,002,177,020,166,254
4923 :002,010,202,208,252,032,253
4929 :077,202,208,240,169,004,197

4935 :005,001,133,001,088,096,139
4941 :172,195,002,044,077,201,000
4947 :048,012,133,254,173,179,114
4953 :002,049,251,005,254,076,214
4959 :099,202,081,251,145,251,100
4965 :200,140,195,002,192,008,070
4971 :208,017,160,064,140,195,123
4977 :002,230,252,044,196,002,071
4983 :016,005,169,007,141,176,121
4989 :002,238,176,002,173,176,124
4995 :002,201,008,096,169,000,095
5001 :141,077,201,076,078,201,143
5007 :169,255,141,077,201,076,038
5013 :078,201,032,115,000,032,095
5019 :040,175,234,234,234,234,026
5025 :234,234,165,071,133,020,250
5031 :165,072,133,021,160,000,206
5037 :177,020,240,213,056,165,020
5043 :020,233,002,133,020,176,251
5049 :002,198,021,177,020,197,032
5055 :069,208,196,200,177,020,037
5061 :197,070,208,189,160,003,000
5067 :177,020,133,251,200,177,137
5073 :020,133,252,169,000,133,148
5079 :253,133,002,133,254,160,126
5085 :000,177,071,170,164,002,037
5091 :177,251,201,018,208,007,065
5097 :169,128,133,253,076,009,233
5103 :203,201,146,208,007,169,149
5109 :000,133,253,076,009,203,151
5115 :041,191,016,002,073,192,254
5121 :005,253,164,254,145,251,049
5127 :230,254,230,002,202,208,109
5133 :211,165,254,160,000,145,180
5139 :071,096,000,000,000,000,186
5145 :000,000,000,000,000,000,025
5151 :000,000,072,020,010,000,133
5157 :153,034,147,154,083,085,181
5163 :080,069,082,066,065,083,232
5169 :073,067,032,066,089,032,152
5175 :077,067,083,079,070,084,003
5181 :032,040,067,041,032,049,066
5187 :057,056,051,034,000,101,110
5193 :020,015,000,129,074,178,233
5199 :049,164,053,048,058,161,100
5205 :065,036,058,139,065,036,228
5211 :178,034,034,167,130,058,180
5217 :137,050,048,000,107,020,203
5223 :016,000,130,000,143,020,156

```

5229 :020,000,153,034,091,067,218
5235 :065,084,065,034,058,144,053
5241 :067,065,084,065,058,144,092
5247 :070,067,079,076,049,052,008
5253 :058,144,066,075,071,068,103
5259 :054,058,162,000,000,032,189
5265 :000,192,165,020,141,158,053

```

```

5271 :203,032,000,192,076,101,243
5277 :199,000,032,014,198,174,006
5283 :158,203,240,003,076,132,207
5289 :200,073,255,049,251,076,049
5295 :134,200,032,057,195,032,057
5301 :066,195,032,218,197,096,217

```

Gli otto brevi programmi danno un esempio delle possibilità del SUPERBASIC: provate a caricarli (con SUPERBASIC attivato) e siamo certi ne sarete entusiasti!

Programma 1

```

5 [FCOL7:[EXTC10:A=32
8 [MXGR33,240,109,5,189,0
10 [FSCRA:[FCOLA+25
12 [CMXVA+100,A
13 [EXTCA+50
15 GETA$:IFA$=""THEN15
16 IFA$="{CLR}"THEN [KMXG:[FCOL4:
  [EXTC7:[BKGD13:A=32:[FSCRA:EN
  D
20 A=ASC(A$):GOTO10

```

Programma 2

```

10 [BMGR1:[CB2K4:[FBMS0:[MGR1:[
  FSCR32
20 [BKGD3:[FCOL4
60 FORC=0TO3:FORX=50TO54:FORY=50
  TO54
70 [MCPLX+C*10,Y+C*10,C:NEXT:NEX
  T:NEXT
75 [BKGD5
80 GETA$:IFA$=""THEN80
90 [CB2K2:[BMGR0:[MGR0:PRINT"
  {CLR}"
100 IFA$<>"Q"THEN10
110 END

```

Programma 3

```

10 [BMGR1:[CB2K4:[FBMS0:[FSCR83
20 [BKGD3:[FCOL4
30 [HRCS53248
60 FORX=0TO319STEP16
61 [DRAW1,0,0,X,199:[DRAW1,319,0
  ,X,199
62 [DRAW1,0,199,X,0:[DRAW1,319,1
  99,X,0
63 NEXT
69 FORY=0TO199STEP16
70 [DRAW1,319,0,0,Y:[DRAW1,0,0,3
  19,Y
71 [DRAW1,0,199,319,Y:[DRAW1,319
  ,199,0,Y

```

```

72 NEXT
74 [DRAW1,0,0,319,199:[DRAW1,0,1
  99,319,0
75 [BKGD5:Z=10
80 GETA$:IFA$=""THEN80
81 [CODEA$:W=ASC(A$):[CHARW,Z,13
  5:[CHRXW,Z,150
85 GETS$:IFS$=""THEN85
90 IFS$="Q"THEN10
92 IFS$=CHR$(13)THEN100
93 Z=Z+9:A$=S$:GOTO81
100 [CB2K2:[BMGR0:PRINT"{CLR}"
110 END

```

Programma 4

```

5 C=1
10 [BMGR1:[CB2K4:[FBMS0:[FSCR99
20 [BKGD3:[FCOL4:[EXTC8
30 X=0:X1=319:Y=0:Y1=199
40 FORK=1TOC+37
61 [DRAWC,X,Y,X1,Y+5:X=X+5:[DRAW
  C,X1,Y,X1-5,Y1:Y=Y+5
62 [DRAWC,X1,Y1,X,Y1-5:X1=X1-5:[
  DRAWC,X,Y1,X+5,Y:Y1=Y1-5
70 NEXT
75 [BKGD5
80 GETA$:IFA$=""THEN80
90 IFC=1THENC=0:GOTO30
100 [CB2K2:[BMGR0:PRINT"{CLR}"
105 [SCRN1,0,2
110 END

```

Programma 5

```

10 [BMGR1:[CB2K4:[FBMS0:[MGR1:[
  FSCR87
20 [BKGD15:[FCOL10:[EXTC1
60 FORC=0TO3:FORX=50TO54:FORY=50
  TO54
70 [MCPLX+C*10,Y+C*10,C:NEXT:NEX
  T:NEXT
71 FORX=0TO319STEP64:FORY=0TO199
  STEP50
72 FORZ=319TO0STEP-64:FORQ=199TO
  0STEP-50

```

```

75 [DRAW1,X,Y,Z,Q
77 NEXT:NEXT:NEXT:NEXT
80 GETA$:IFA$=""THEN80
90 [CB2K2:[BMGR0:[MCGR0:PRINT"
  {CLR}"
110 END

```

Programma 6

```

5 [FCOL7:[EXTC10:A=32
8 [MXGR22,239,109,255,189,0
10 [FSCRA:[FCOLA+25
12 [CMXVA+100,A
13 [EXTCA+50
15 GETA$:IFA$=""THEN15
16 IFA$="{CLR}"THEN[KMXG:[FCOL4:
  [EXTC7:[BKGD13:A=32:[FSCRA:EN
  D
20 A=ASC(A$):GOTO10

```

Programma 7

```

5 C=1
10 [BMGR1:[CB2K4:[FBMS0:[FSCR99
20 [BKGD3:[FCOL4:[EXTC8
30 X=0:X1=319:Y=0:Y1=199
40 FORK=1TOC+37
61 [DRAWC,X1,Y,X,Y+4:X=X+1:[DRAW
  C,Y1,X,Y+5,Y1:Y=Y+3
62 [DRAWC,X1,Y1,X,Y1-5:X1=X1-5:[
  DRAWC,X,Y1,X+5,Y:Y1=Y1-5
70 NEXT
75 [BKGD5
80 GETA$:IFA$=""THEN80
90 IFC=1THENC=0:GOTO30

```



```

100 [CB2K2:[BMGR0:PRINT"{CLR}"
105 [SCRN1,3,2
110 END

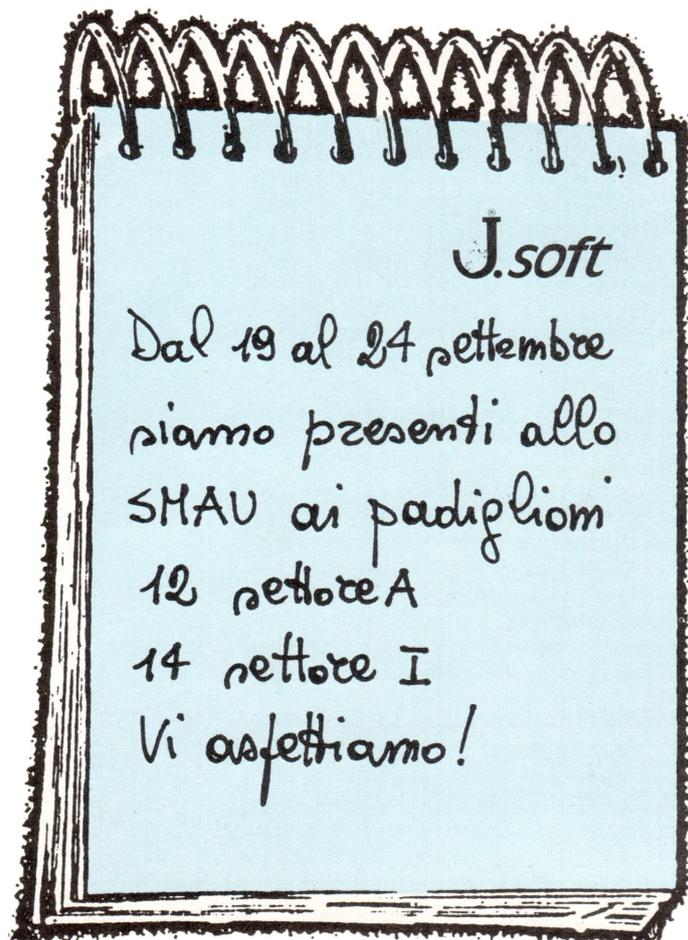
```

Programma 8

```

5 X=170:Y=100:R=99
10 [BMGR1:[CB2K4:[FBMS0:[FSCR99
20 FORA=0TO/2STEP/180
30 C=COS(RA)*R:S=SIN(RA)*R
40 [FLIPX+C,Y+S:[FLIPX+C,Y-S:[FL
  IPX-C,Y+S:[FLIPX-C,Y-S:NEXT
50 [PLOTX,Y
55 IFQ=0THENQ=1:X=137:Y=109:R=78
  :GOTO20
60 GETA$:IFA$=""THEN60
70 [BMGR0:[CB2K2:PRINT"{CLR}"

```





Duello spaziale

di A. Hayes
trad. e adatt. di F. Stella

Questo è un gioco d'azione per due persone per il VIC 20 ed il C64; prevede l'uso delle paddle, e la versione per il C64 è stata trascritta completamente in linguaggio macchina.

Uno dei problemi incontrati nella programmazione dei giochi con il VIC 20, è la mancanza della seconda porta di controllo per joystick (che il C64 invece, ha). Per questo motivo la maggior parte dei giochi sono stati scritti per un solo giocatore. Ma non c'è dubbio che il divertimento di un gioco a due sia maggiore, in quanto avere come avversario il computer ci pone in condizione di svantaggio ogni qualvolta siano in gioco i tempi di riflesso e le deduzioni logico-matematiche; così, se non si provvede a limitarne le capacità, "lui" vince sempre. In un gioco a due, invece, si affronta un altro essere umano che ha la stessa nostra debolezza e discontinuità di rendimento. Non solo c'è dunque una più alta probabilità di vincere, ma il gioco a due permette di usare il computer insieme, nello stesso momento. Vi sono tre modi per programmare giochi a due con azione simultanea per il VIC 20: una seconda porta per joystick può essere creata con una apposita interfaccia alla USER PORT (non facile per principianti); si può usare la tastiera per controllare il gioco (ma risulta macchinoso, in due) oppure il gioco può essere scritto per l'uso delle paddle; questi accessori si trovano in commercio a coppie, collegati allo stesso connettore per joystick (nel C64 andranno collegati alla porta 1).

Animazione in alta risoluzione

Il gioco DUELLO SPAZIALE dà, ad ogni giocatore, il comando di un caccia stellare, situato ai margini dello schermo, mobile (attraverso opportune rotazioni della manopola) in senso verticale.

Provando ad osservare da vicino la nave spaziale e ruotando lentamente la

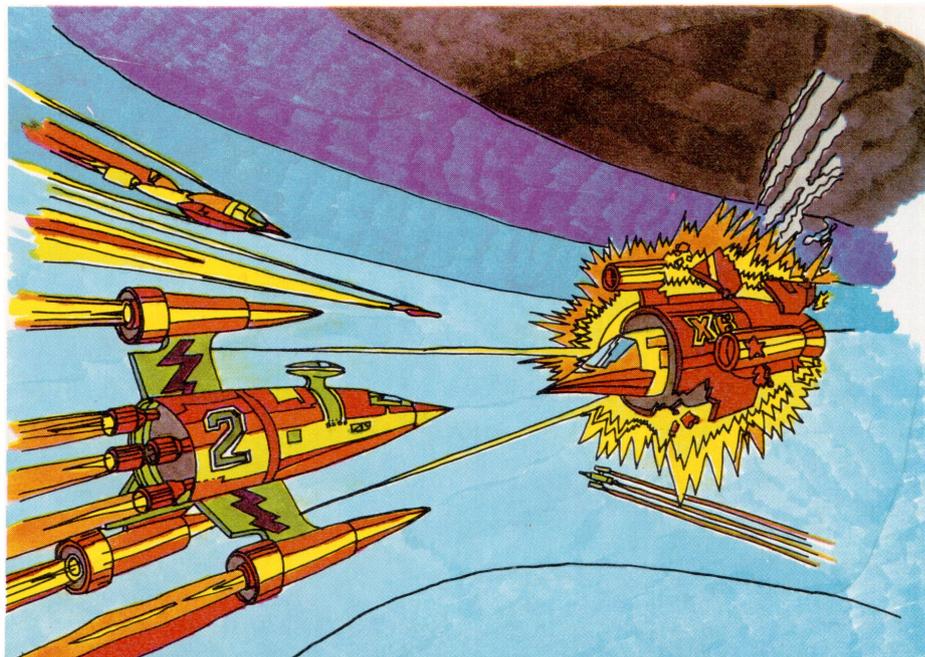
manopola, si può notare che, a differenza di molti giochi per il VIC 20 ed il C64, gli oggetti non si muovono in grossolani incrementi dello spazio di un carattere ma scorrono dolcemente su e giù di un pixel per volta.

Questo tipo di risoluzione sarebbe incredibilmente lenta se programmata in BASIC: la versione VIC usa una subroutine in LM per ottenere questo fine movimento, la versione C64 usa gli sprite ed è interamente scritta in LM.

Ruotando la manopola più velocemente, si scopre un altro tipo di movimento: balzi veloci in un senso o nell'altro, questo perché lo stato delle paddle è rappresentato nella memoria sotto forma di un valore assoluto e non direzionale, come nel caso dei joystick. Con un rapido movimento del polso è possibile variare discretamente il valore, prima della successiva lettura, saltando così da una posizione all'altra senza passare per le intermedie. Entrambi i movimenti si rivelano molto utili nel gioco: balzi per evitare il fuoco nemico e piccoli ed accurati spostamenti per mirare con precisione.

Duello tra navi spaziali

Lo scopo di questo gioco è semplice: colpire la nave avversaria più di quanto non si sia colpiti da questa. Per azionare il FUOCO: un raggio sibila immediatamente attraverso lo schermo (a velocità di LM) verso il bersaglio; se il colpo va a segno, si aziona un suono di esplosione e lampi colorati appaiono sullo schermo. Durante il gioco bisogna, ovviamente, evitare i colpi nemici oltre che sparare; l'incrociarsi di colpi può divenire così rapido che solo i giocatori più scaltri riescono a stare al passo. Ogni colpo mandato a segno sulla nave nemica vale dieci punti; mentre, per prevenire l'uso di raffiche sconsiderate, ogni emissione laser detrae un punto, così ogni centro fa guadagnare in realtà nove punti. I punteggi aggiornati sono evidenziati nella parte alta dello schermo ed il gioco finisce quando uno dei giocatori raggiunge 80 punti per il VIC 20 e 500 punti per il C64; per riprendere il gioco ed azzerare il punteggio basterà premere il tasto delle paddle oppure ri-



spondere da tastiera alla richiesta del computer.

Suggerimento: nella versione VIC, può talvolta succedere che le paddle funzionino in modo irregolare, in tal caso premere RUN/STOP e RESTORE e ripartire, questo pulisce la memoria eli-

minando alcuni valori che possono interferire con il controllo delle paddle; ricordare inoltre di non lasciare premuto il tasto del registratore perché questo impedisce l'uso regolare della paddle sinistra. Per caricare la versione C64 (tutta in LM), bisogna usare il

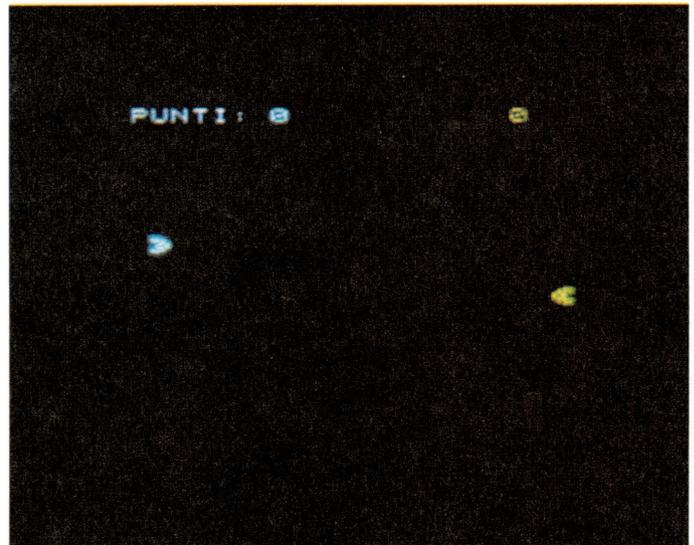
programma MLX, che facilita enormemente il lavoro rendendolo praticamente esente da errori. Le informazioni necessarie per l'uso di MLX sono: indirizzo di partenza: 49152, indirizzo finale: 50393, per avviare il gioco digitale SYS49152.

Duello spaziale versione per VIC 20

```

0 PRINT"{CLR}":FORI=6656TO6891:READA:POKE
  I,A:NEXT :rem 188
1 GOSUB200:PRINT"{CLR}" :rem 225
3 POKE4,0:POKE2,0:POKE3,0 :rem 225
4 FORI=7167TO7600:POKEI,0:NEXTI:POKE36869
  ,255 :rem 247
7 FORI=1TO8:POKE7662,0:NEXT :rem 195
8 A$="{ 2 DES}{PUR}22222222222222222222
  { 2 DES}" :rem 178
9 C$="{ 2 DES}{PUR}33333333333333333333
  { 2 DES}" :rem 199
10 B$="{HOME}{ 22 GIU'}" :rem 209
11 D$="{ 2 DES}{BLK}{ 18 SPAZI}{ 2 DES}":
  POKE36879,8:F=80 :rem 82
30 FORI=7747TO8099STEP22:POKEI,J:J=J+1:NE
  XT :rem 206
35 FORI=7747+30720TO30720+8099STEP22:POKE
  I,3:NEXT :rem 163
40 J=33:FORI=7766TO8118STEP22:POKEI,J:J=J
  +1:NEXT :rem 239
45 FORI=7766+30720TO8118+30720STEP22:POKE
  I,7:NEXT :rem 161
46 PRINT"{HOME}{RVS}{WHT}PUNTI:" :rem 47
50 SYS6656 :rem 59
52 PRINT"{HOME}{ 6 DES}{RVS}{CYN}"PEEK(2)
  "{SIN}";TAB(17);"{YEL}"PEEK(3)"{SIN}"
  :rem 165
54 FORT=1TO30:NEXT :rem 145
55 SYS6811 :rem 57
56 ONPEEK(4)GOSUB80 :rem 154
57 SYS6864:ONPEEK(5)GOSUB100 :rem 214
58 SYS6772 :rem 66
60 GOTO50 :rem 4
80 A=PEEK(0)+33+3 :rem 123
82 B=INT(A/8)-1:D=(A-B*8)+2 :rem 166
84 POKE7566+D,255 :rem 170
90 IFD<7THENPRINTLEFT$(B$,B);A$:SYS6832:P
  RINTLEFT$(B$,B);D$:POKE7566+D,0
  :rem 117
91 H=PEEK(2)-1:IFH>-1THENPOKE2,H:GOTO93
  :rem 133
92 H=0 :rem 32
93 IFD>7THENPRINTLEFT$(B$,B+1);C$:SYS6832
  :PRINTLEFT$(B$,B+1);D$:POKE7566+D,0
  :rem 52
94 IFPEEK(7432+PEEK(0)+3)<>0THENGOSUB120
  :rem 93
95 RETURN :rem 78
100 A=PEEK(1)+33+3 :rem 165
102 B=INT(A/8)-1:D=(A-B*8)+2 :rem 207
103 POKE7566+D,255 :rem 210
104 H=PEEK(3)-1:IFH>-1THENPOKE3,H:GOTO106
  :rem 221
105 H=0 :rem 75
106 IFD<7THENPRINTLEFT$(B$,B);A$:SYS6832:
  PRINTLEFT$(B$,B);D$:POKE7566+D,0
  :rem 163
107 IFD>7THENPRINTLEFT$(B$,B+1);C$:SYS683
  2:PRINTLEFT$(B$,B+1);D$:POKE7566+D,0

```



```

:rem 96
109 IFPEEK(7168+PEEK(1)+3)<>0THENGOSUB140
  :rem 147
110 RETURN :rem 114
120 POKE36878,15 :rem 100
121 FORI=255TO130STEP-2:POKE36877,I:POKE3
  6879,INT(RND(1)*7)+8:NEXT :rem 225
122 POKE2,PEEK(2)+10 :rem 86
123 IFPEEK(2)=>FTHEN150 :rem 83
124 POKE36879,8:POKE36877,0:RETURN
  :rem 41
140 POKE36878,15 :rem 102
141 FORI=255TO130STEP-2:POKE36877,I:POKE3
  6879,INT(RND(1)*7)+8:NEXT :rem 227
142 POKE3,PEEK(3)+10 :rem 90
143 IFPEEK(3)=>FTHEN150 :rem 86
144 POKE36879,8:POKE36877,0:RETURN
  :rem 43
150 PRINT"{HOME}{RVS}{WHT}PUNTI:"
  :rem 91
152 PRINT"{HOME}{ 6 DES}{RVS}{CYN}"PEEK(2)
  );TAB(18);"{YEL}"PEEK(3):POKE36877,0
  :rem 233
154 POKE36879,8 :rem 62
156 PRINT"{HOME}{ 2 GIU' }{RVS}{GRN}GAME O
  VER!" :rem 73
157 POKE198,0:WAIT198,1:RUN :rem 103
200 POKE36879,8:POKE36869,240:PRINT"{CLR}
  " :rem 15
210 PRINT"{RVS}{YEL}{ 22 SPAZI}{OFF}
  { 4 SPAZI}{CYN}DUELLO SPAZIALE{YEL}
  { 3 SPAZI}{OFF}"; :rem 206
220 PRINT"{RVS}{YEL}{ 22 SPAZI}":rem 21
240 PRINT"{ 4 GIU' }{BLU}{ 4 SPAZI}PREMI U
  N TASTO :rem 83
250 POKE198,0:WAIT198,1:RETURN :rem 76
260 DATA173,8,144,74,133,0,234,170,169,24
  0,157,0,28,232,169,252,157,0,28,232

```

```

                :rem 208
262 DATA169,14,157                :rem 174
265 DATA0,28,232,169,59,157,0,28,232,169,
    59,157,0,28,232,169,14,157,0,28,232,1
    69,252                            :rem 116
270 DATA157,0,28,232,169,240,157,0,28,173
    ,9,144,74,133,1,170,169,15,234,157,8,
    29                                :rem 161
275 DATA232,169                    :rem 27
280 DATA63,157,8,29,232,169,112,157,8,29,
    232,169,220,157,8,29,232,169,220,157,
    8,29,232                            :rem 221
290 DATA169,112,157,8,29,232,169,63,157,8
    ,29,232,169,15,157,8,29,96,251,4,24,6
    6,252                                :rem 84
300 DATA166,0,169,0,160,0,157,0,28,232,20
    0,192,9,208,247,166,1,169,0,160,0
                :rem 82
305 DATA157,8,29,232                :rem 13
310 DATA200,192,9,208,247,96,3,255,179,25
    2,81,119,1,111,174,17,145,224,110,240
    ,5,169,0                            :rem 183
320 DATA133,4,96,169,1,133,4,96,253,162,2
    24,56,169,14,141,14,144,162,255,142
                :rem 211
325 DATA11,144,202                  :rem 155
330 DATA160,0,200,192,74,208,251,224,128,
    208,241,169,0,141,11,144,96,119,111,1
    59                                    :rem 135
335 DATA163,169                    :rem 27
340 DATA127,141,34,145,174,32,145,169,255
    ,141,34,145,224,119,240,5,169,0,133
                :rem 205
345 DATA5,96,169,1                  :rem 175
350 DATA133,5,96,219                :rem 13

```

Duello spaziale versione per C64

```

49152 :169,055,141,000,208,169,230
49158 :002,141,016,208,169,032,062
49164 :141,002,208,032,070,196,149
49170 :169,147,032,210,255,169,232
49176 :012,141,033,208,169,192,011
49182 :141,248,007,169,193,141,161
49188 :249,007,169,193,141,249,020
49194 :007,169,003,141,021,208,079
49200 :169,000,141,134,002,169,151
49206 :000,141,039,208,169,002,101
49212 :141,040,208,169,000,160,010
49218 :000,145,000,200,192,024,115
49224 :208,249,169,015,141,024,110
49230 :212,169,017,141,005,212,066
49236 :169,246,141,006,212,169,003
49242 :050,141,000,212,141,001,123
49248 :212,169,000,160,000,153,022
49254 :060,003,200,192,060,208,057
49260 :248,169,000,170,168,024,119
49266 :109,025,212,144,004,200,040
49272 :140,060,003,202,208,243,208
49278 :169,000,170,168,024,109,254
49284 :026,212,144,004,200,140,090
49290 :062,003,202,208,243,169,001

```

```

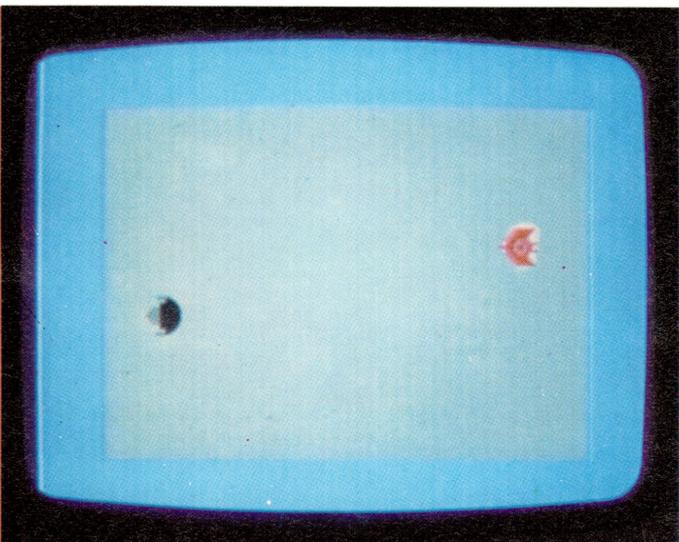
49296 :000,205,060,003,240,007,147
49302 :144,005,169,000,141,060,157
49308 :003,169,180,205,060,003,008
49314 :176,005,169,180,141,060,125
49320 :003,169,000,205,062,003,098
49326 :144,005,169,000,141,062,183
49332 :003,169,180,205,062,003,034
49338 :176,005,169,180,141,062,151
49344 :003,024,173,060,003,105,048
49350 :050,141,001,208,024,173,027
49356 :062,003,105,050,141,003,056
49362 :208,056,173,060,003,074,016
49368 :074,074,024,105,001,141,123
49374 :090,003,056,173,062,003,097
49380 :074,074,074,024,105,001,068
49386 :141,092,003,169,000,141,012
49392 :091,003,173,090,003,010,098
49398 :046,091,003,010,046,091,021
49404 :003,010,141,072,003,173,142
49410 :091,003,141,073,003,173,230
49416 :072,003,010,046,091,003,233
49422 :010,046,091,003,024,109,041
49428 :072,003,141,090,003,173,246
49434 :091,003,109,073,003,141,190
49440 :091,003,024,173,090,003,160
49446 :105,005,141,090,003,173,043
49452 :091,003,105,004,141,091,223
49458 :003,169,000,141,093,003,203
49464 :173,092,003,010,046,093,217
49470 :003,010,046,093,003,010,227
49476 :141,074,003,173,093,003,043
49482 :141,075,003,173,074,003,031
49488 :010,046,093,003,010,046,032
49494 :093,003,024,109,074,003,136
49500 :141,092,003,173,093,003,085
49506 :109,075,003,141,093,003,010
49512 :024,173,092,003,105,005,250
49518 :141,092,003,173,093,003,103
49524 :105,004,141,093,003,173,123
49530 :090,003,133,251,173,091,095
49536 :003,133,252,173,092,003,016
49542 :133,253,173,093,003,133,154
49548 :254,173,001,220,041,004,065
49554 :240,003,076,077,194,160,128
49560 :000,169,067,145,251,024,040
49566 :165,252,105,212,133,252,253
49572 :169,002,145,251,056,165,184
49578 :252,233,212,133,252,200,172
49584 :192,030,208,229,056,165,032
49590 :251,233,080,141,064,003,186
49596 :165,252,233,000,141,065,020
49602 :003,024,165,251,105,040,014
49608 :141,066,003,165,252,105,164
49614 :000,141,067,003,056,173,134
49620 :064,003,229,253,141,068,202
49626 :003,173,065,003,229,254,177
49632 :013,068,003,176,061,056,089
49638 :173,066,003,229,253,141,071
49644 :068,003,173,067,003,229,011
49650 :254,013,068,003,144,042,254

```

49656 :032,121,195,024,173,078,103
 49662 :003,105,011,141,078,003,083
 49668 :173,079,003,105,000,141,249
 49674 :079,003,056,173,078,003,146
 49680 :233,244,141,068,003,173,110
 49686 :079,003,233,001,013,068,163
 49692 :003,144,003,076,015,195,208
 49698 :162,255,160,015,136,208,202
 49704 :253,202,208,248,173,078,178
 49710 :003,013,079,003,240,014,142
 49716 :173,078,003,208,003,206,211
 49722 :079,003,206,078,003,032,203
 49728 :149,195,160,030,169,032,031
 49734 :145,251,136,192,000,016,042
 49740 :247,173,001,220,041,008,254
 49746 :240,003,076,012,195,160,000
 49752 :000,169,067,145,253,024,234
 49758 :165,254,105,212,133,254,193
 49764 :169,006,145,253,056,165,126
 49770 :254,233,212,133,254,200,112
 49776 :192,030,208,229,056,165,224
 49782 :253,233,080,141,064,003,124
 49788 :165,254,233,000,141,065,214
 49794 :003,024,165,253,105,040,208
 49800 :141,066,003,165,254,105,102
 49806 :000,141,067,003,056,173,070
 49812 :064,003,229,251,141,068,136
 49818 :003,173,065,003,229,252,111
 49824 :013,068,003,176,060,056,024
 49830 :173,066,003,229,251,141,005
 49836 :068,003,173,067,003,229,203
 49842 :252,013,068,003,144,041,187
 49848 :032,121,195,173,076,003,016
 49854 :105,010,141,076,003,173,186
 49860 :077,003,105,000,141,077,087
 49866 :003,056,173,076,003,233,234
 49872 :244,141,084,003,173,077,162
 49878 :003,233,001,013,084,003,039
 49884 :144,003,076,068,195,162,100
 49890 :255,160,015,136,208,253,229
 49896 :202,208,248,173,076,003,118
 49902 :013,077,003,240,014,173,246



49908 :076,003,208,003,206,077,049
 49914 :003,206,076,003,032,149,207
 49920 :195,160,030,169,032,145,219
 49926 :253,136,192,000,016,247,082
 49932 :076,109,192,169,147,032,225
 49938 :210,255,169,000,141,021,046
 49944 :208,160,012,162,010,032,096
 49950 :240,255,160,000,185,236,082
 49956 :195,200,032,210,255,192,096
 49962 :044,208,245,165,197,201,078
 49968 :064,240,250,165,197,201,141
 49974 :025,240,008,165,197,201,122
 49980 :039,208,238,096,096,076,045
 49986 :000,192,169,147,032,210,048
 49992 :255,169,000,141,021,208,098
 49998 :160,012,162,010,032,240,182
 50004 :255,160,000,185,025,196,137
 50010 :200,032,210,255,192,044,255
 50016 :208,245,165,197,201,064,152
 50022 :240,250,165,197,201,025,156
 50028 :240,008,165,197,201,039,190
 50034 :208,238,096,096,076,000,060
 50040 :192,162,255,238,032,208,183
 50046 :202,208,250,169,129,141,201
 50052 :004,212,162,255,160,115,016
 50058 :136,208,253,202,208,248,113
 50064 :169,128,141,004,212,160,190
 50070 :011,162,000,024,032,240,107
 50076 :255,169,032,032,210,255,085
 50082 :160,012,162,000,024,032,040
 50088 :240,255,169,032,032,210,082
 50094 :255,160,026,162,000,024,033
 50100 :032,240,255,169,032,032,172
 50106 :210,255,160,027,162,000,232
 50112 :024,032,240,255,169,032,176
 50118 :032,210,255,160,010,162,003
 50124 :000,024,032,240,255,174,161
 50130 :078,003,173,079,003,032,066
 50136 :205,189,160,025,162,000,189
 50142 :024,032,240,255,174,076,255
 50148 :003,173,077,003,032,205,209
 50154 :189,096,080,076,065,089,061
 50160 :069,082,032,049,032,087,079
 50166 :073,078,083,013,013,032,026
 50172 :032,032,032,032,032,032,188
 50178 :032,032,072,073,084,032,071
 50184 :089,032,084,079,032,080,148



50190 :076,065,089,032,065,071,156
 50196 :065,073,078,032,032,080,124
 50202 :076,065,089,069,082,032,183
 50208 :050,032,087,073,078,083,179
 50214 :013,013,032,032,032,032,192
 50220 :032,032,032,032,032,072,020
 50226 :073,084,032,089,032,084,188
 50232 :079,032,080,076,065,089,221
 50238 :032,065,071,065,073,078,190
 50244 :032,032,160,000,185,084,049
 50250 :196,153,000,048,200,192,095
 50256 :147,208,245,096,001,255,008
 50262 :000,000,003,192,000,003,028
 50268 :240,000,063,248,000,063,194
 50274 :252,000,063,254,015,255,169
 50280 :254,000,063,255,063,255,226
 50286 :255,000,063,255,255,255,169
 50292 :255,000,063,255,063,255,239
 50298 :255,000,063,255,015,255,197
 50304 :254,000,063,254,000,063,250
 50310 :252,000,063,248,000,003,188
 50316 :240,000,003,192,000,255,062



Duello Spaziale

50322 :000,255,000,127,252,000,012
 50328 :015,248,001,255,240,000,143
 50334 :063,224,007,255,192,000,131
 50340 :255,128,031,245,192,003,250
 50346 :228,224,007,213,127,015,216
 50352 :142,048,255,255,240,015,107
 50358 :142,048,007,213,127,003,210
 50364 :228,224,031,245,192,000,084
 50370 :255,128,007,255,192,000,007
 50376 :063,224,001,255,240,000,215
 50382 :015,248,000,127,252,255,079
 50388 :255,013,013,013,013,137,144

IN MILANO - VIA MASCHERONI, 14

IL VOSTRO “NUOVISSIMO” COMPUTER SHOP

Libri e riviste di elettronica e informatica.

CORSI CONTINUI TUTTO L'ANNO CON I MIGLIORI SOFTERISTI



IL TELEFONO È 02-437.385

Il Signore dei draghi

di D. Berdon

trad. ed adatt. di U.G. Barzaghi

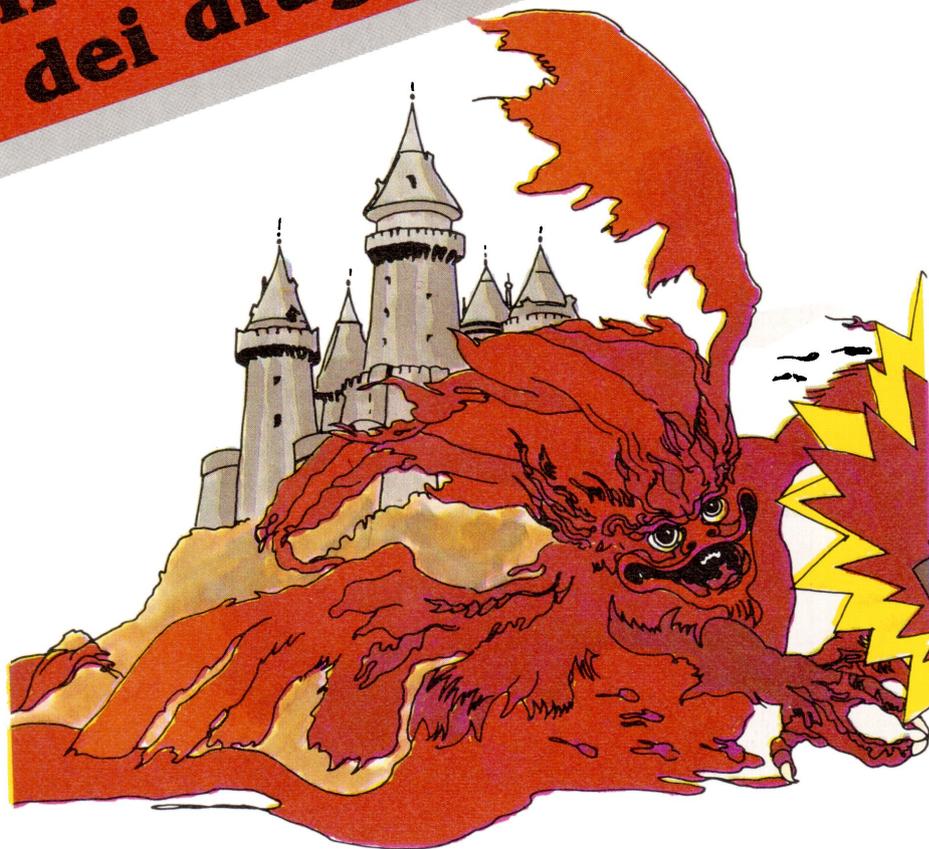
La vostra missione, in questo gioco di avventura e prontezza di riflessi, consiste nel muoversi attraverso la contrada fino a raggiungere il castello ed a salvare la principessa - se ne siete capaci. Ci sono dragoni di ogni tipo da sottomettere, ed un malvagio, subdolo mago che non smette di inseguirvi. Il gioco ha quattro livelli di difficoltà. Originariamente scritto per il VIC privo di espansione di memoria, è stato adattato al C64. È necessario un joystick.

Questo gioco coinvolge tre coraggiosi cavalieri, tornati dalla guerra giusto in tempo per scoprire che la contrada circostante il loro castello è infestata da dragoni piazzati lì da un malvagio stregone. Una bellissima principessa è intrappolata nel castello, ed attende ansiosamente il primo cavaliere che riuscirà a salvarla. Uno alla volta i cavalieri tentano di uccidere tutti i dragoni ed entrare nel castello per salvare la principessa.

I cavalieri scoprono ben presto come la caccia presenti molte difficoltà e pericoli. Un dragone divora chiunque si avvicini alla sua testa. L'unico modo di liberarsi di uno di loro consiste nel colpirla allo stomaco. A guardia del castello è posto un dragone fantasma che può essere eliminato solo quando tutti gli altri sono stati sopraffatti, (il ponte levatoio del castello rimarrà alzato finché esso non sarà sconfitto). A volte un dragone apparirà all'improvviso, e talvolta un dragone di cui pensavate esservi liberati riappare per cercare un'altra volta di divorare il cavaliere. E infine, ma certamente non il minore dei pericoli, lo stesso malvagio stregone insegue i cavalieri mentre essi sono a caccia dei draghi. Se il mago raggiunge il cavaliere, questo viene ucciso.

Come si gioca

Muovete il vostro cavaliere per la contrada con il joystick. Disponete di tre cavalieri al via, ma fate attenzione, essi



vengono rapidamente divorati dai draghi...

Quando affrontate un dragone, dovete agire rapidamente. Usando il vostro pulsante di sparo, mirate allo stomaco del drago. Quello è il loro solo punto vulnerabile.

La vostra unica difesa nei confronti dello stregone malvagio è la velocità. Dovete dare costantemente attenzione a quanto dista da voi. Ricordatevi: se vi raggiunge, il vostro cavaliere è perduto.

"Il Signore dei draghi" (versione VIC 20)

La versione originale de "Il Signore dei draghi", realizzata per il VIC-20, utilizza un set di caratteri modificato da programma. Poiché sono necessari ben 232 valori numerici per creare i 29 caratteri speciali utilizzati in questo gioco, è stato necessario creare una registrazione dei dati contenente i numeri e leggerla all'inizio del programma per costruire i caratteri ridefiniti.

Il programma 1 è il programma principale che controlla il gioco. Il programma 2 è la registrazione dei dati necessari per definire il set di caratteri utilizzato dal gioco, (Vi preghiamo di notare che, a causa della necessità di proteg-

gere un blocco di memoria riservato al set di caratteri, la versione VIC funzionerà esclusivamente su un VIC privo di espansione di memoria. Siete quindi pregati di togliere eventuali cartucce di espansione RAM).

Battete attentamente il Programma 1, controllatene la correttezza, e registratelo su nastro. Se mandate in esecuzione il programma 1 mediante una istruzione RUN, cercherà di caricare i dati registrati che non sono ancora stati creati. Quindi, battete NEW e copiate il programma 2. Fate attenzione nel battere questi valori numerici, poiché ogni errore provocherà caratteri mal formati. Quando siete certi che ogni cosa è a posto, mandate in esecuzione il programma 2. Il VIC vi chiederà di premere i tasti Record e Play prima di cominciare a scrivere i dati su nastro subito dopo la registrazione del programma 1. Dopo circa un minuto sullo schermo apparirà la scritta "REGISTRAZIONE NASTRO CREATA" ed il nastro si arresterà. È consigliabile salvare una copia del programma 2 su di un altro nastro, così che, nel caso in cui fosse nuovamente necessario, non dobbiate ricopiare tutti quei numeri. Riavvolgete il nastro completamente e battete LOAD per caricare il primo

programma. Quando il programma è stato caricato, battete RUN.

Livelli di difficoltà

Inizialmente appariranno sullo schermo le istruzioni. Vi si chiederà di scegliere tra quattro livelli di difficoltà premendo un opportuno tasto di funzione. Il grado di difficoltà è dato dalla velocità con cui il mago insegue il cavaliere. Il livello FACILE (F1) rappresenta una vera sfida per i soli principianti. Potete aspettarvi di essere battuti abbastanza spesso mentre imparate a conoscere i punti critici del gioco. I 2 livelli successivi, NORMALE (F3) e DIFFICILE (F5), hanno difficoltà crescenti. Il livello MOLTO DIFFICILE (F7) è riservato agli esperti. Dovete essere particolarmente pronti di riflessi ed abili nell'utilizzare il joystick per riuscire ad entrare nel castello a questo livello.

"Il Signore dei draghi" (versione C64)

la versione C64 de "Il signore dei draghi", funziona, suppergiù, come la versione VIC. Tuttavia, i colori sono in parte differenti. Vi accorgerete si essere rappresentati da una figura lampeggiante in diverse tonalità di grigio, un effetto non implementabile sul VIC a causa della limitata capacità di memoria. Lo stregone che vi insegue lampeggia con tutti i colori dell'arcobaleno (magico!) e, tramite i suoi poteri soprannaturali, vi insegue dimezzando costantemente la distanza che vi separa. Se, in qualsiasi momento, il suo balzo lo porta ad una unità video di distanza da voi, siete spacciati.

I dragoni rappresentano il principale ostacolo alla vostra cavalleresca avventura. Per eliminare un drago, dovette porvi direttamente davanti al suo stomaco e premere il pulsante di sparò: il dragone scomparirà. Il drago fantasma posto a guardia del ponte levatoio deve essere eliminato per ultimo.

Vi accorgerete che i draghi si spostano qui e là ed a volte cambiano colore, ma si tratta di semplice irrequietezza. Per di più un dragone apparirà occasionalmente dal nulla per mettere alla prova il vostro coraggio.

Le tecniche di programmazione utilizzate in questa versione, così come nella versione VIC, sono piuttosto immediate. Vengono utilizzati caratteri ridefiniti per i dragoni, il castello, i cavalieri ed il mago. I dati necessari sono immagazzinati nelle locazioni di memoria da 14336 a 16384, anche se solo una parte di essi viene utilizzata. I caratteri vengono inseriti sullo schermo tramite istruzioni POKE, così come il loro colore (le locazioni di colore dello schermo sul C64 vanno da 55296 a 56295). Semplici effetti sonori vengono creati utilizzando due voci del chip SID. La prima voce controlla i dragoni ed i cavalieri, così come i suoni del ponte levatoio; la voce due genera i suoni associati al movimento.

Il Signore dei draghi versione per VIC 20

Programma 1

```

100 PRINT"{CLR}":IFPEEK(7448)=60ANDPEEK(7
    679)=160THEN125 :rem 111
105 PRINT" STO LEGGENDO I DATI" :rem 13
110 OPEN1,1,0,"DATI SU NASTRO" :rem 191
115 FORD=7448TO7679:INPUT#1,A:POKED,A:NEX
    T :rem 116
120 CLOSE1:POKE56,29 :rem 165
125 CLR:PRINT"{CLR}{BLK}":U=36878:DR=15:K
    =2:JS=37154:P=7712:SW=0:S1=U-2:GOSUB2
    00:GOSUB280 :rem 58
130 ZZ=7912:POKEZZ,35 :rem 9
135 POKEJS,127:V=PEEK(JS-2)AND128
    :rem 216
140 E--(V=0):POKEJS-2,255:V=PEEK(JS-3)
    :rem 197
145 S--((VAND8)=0):W--((VAND16)=0):N--((V
    AND4)=0):SW--((VAND32)=0):Q=P
    :rem 194
150 IFSANDPEEK(P+22)=32THENP=P+22
    :rem 221
155 IFWANDPEEK(P-1)=32THENP=P-1 :rem 132
160 IFNANDPEEK(P-22)=32THENP=P-22
    :rem 221
165 IFEANDPEEK(P+1)=32THENP=P+1 :rem 111
170 POKEP+30720,0 :rem 154
175 GOSUB420:IFQ<>PTHENPOKEQ,32:POKEP,53
    :rem 36
180 IFSWTHENGOSUB375 :rem 32
185 IFPEEK(P+1)=37ORPEEK(P+1)=60ORPEEK(P-
    1)=42THENGOSUB330 :rem 9
190 IFPEEK(P+22)=37ORPEEK(P+22)=60ORPEEK(
    P+22)=42THENGOSUB330 :rem 156
195 GOSUB470:GOTO135 :rem 198
200 PRINT"{GIU'}IL SIGNORE DEI DRAGHI"
    :rem 161
205 PRINT"{GIU'}OBIETTIVO: UCCIDERE":PRIN
    T"TUTTI I DRAGHI ED EN-" :rem 146
210 PRINT"TRARE NEL CASTELLO PER":PRINT"
    {SU}SPOSARE LA PRINCIPESSA" :rem 134
215 PRINT"{GIU'}ELIMINA I DRAGHI MET-":PR
    INT"TENDOTICI DI FRONTE E" :rem 210
217 PRINT"PREMENDO IL PULSANTE":PRINT"DI
    SPARO." :rem 17
220 PRINT"{GIU'}PER ULTIMO DEVI ELIMI-":P
    RINT"{SU}NARE IL DRAGO FANTASMA"
    :rem 11
225 PRINT:PRINT"PREMI: F1-FACILE":PRINTTA
    B(7)"F3-NORMALE" :rem 78
230 PRINTTAB(7)"F5-DIFFICILE":PRINTTAB(7)
    "F7-IMPOSSIBILE" :rem 64
235 SL=15:GETA$:IFA$=""THEN235 :rem 211
240 IFA$<CHR$(134)ORA$>CHR$(136)THENPRINT
    "{CLR}":RETURN :rem 141
245 IFA$=CHR$(134)THENSL=10:PRINT"{CLR}":
    RETURN :rem 218
250 IFA$=CHR$(135)THENSL=6:PRINT"{CLR}":R
    ETURN :rem 172
255 SL=3:PRINT"{CLR}":RETURN :rem 99
260 CL=INT(RND(1)*2)*2 :rem 190
265 CS=L+30720:POKECS,CL:POKECS+1,CL:POKE
    CS+22,CL:POKECS+23,CL :rem 70
270 POKEL,M:POKEL+1,M+1:POKEL+22,M+2:POKE
    L+23,M+3:RETURN :rem 195
275 POKEL,32:POKEL+1,32:POKEL+22,32:POKEL
    +23,32:RETURN :rem 17
280 FORI=1TO126:PRINT"{ 4 SPAZI}";:NEXT:P
    RINT" {HOME}":POKE38905,0 :rem 31
285 POKEU+1,30:FORC=7424TO7431:POKEC,0:NE
    XT:POKEU-9,255 :rem 95
290 L=7888:M=45:CL=7:GOSUB265:POKE38629,C
    L:POKE7909,49:CL=4:L=L-3:M=60:GOSUB26
    5:F=36 :rem 31
295 FORX=7680TO7701:POKEX,F:NEXT:FORX=770
    2TO8142STEP22:POKEX,F:POKEX+21,F:NEXT
    :rem 20
300 FORX=8164TO8185:POKEX,F:NEXT:FORX=1TO
    15:Z=INT(RND(1)*2) :rem 198
305 READA:M=41:L=7726+A:IFZTHENM=37
    :rem 105
310 GOSUB260:NEXT:POKEU,15:POKE7712,53:PO
    KE7693,53:POKE7694,53 :rem 240
315 DATA1,15,31,69,101,111,199,206,212,26
    8,275,316,353,362,367 :rem 40
320 FORX=49TO52:POKE7909,X:POKES1,(X-44)*

```

```

28: FORT=1TO1000:NEXT:NEXT:GOSUB325:RE
TURN :rem 224
325 POKEU,0:POKES1,0:RETURN :rem 221
330 POKEU,15:FORF=1TO2:FORX=130TO200:POKE
S1,X:NEXT :rem 191
335 FORX=200TO130STEP-1:POKES1,X:NEXT:POK
EP,F+53:NEXT:GOSUB325 :rem 239
340 POKE7695-K,36:K=K-1:IFK=-1THEN455
:rem 149
345 POKEP,32:P=7712:POKEP,53:RETURN
:rem 171
350 L=P-21:M=56:CL=0:GOSUB265:GOSUB400:L=
P-21:GOSUB275:POKEU,15 :rem 86
355 FORX=52TO49STEP-1:POKE7909,X:POKES1,(
X-44)*28:FORT=1TO1000:NEXT:NEXT:GOSUB
325 :rem 104
360 FORX=7907TO7909:POKEX-1,32:POKEX+3072
0,0:POKEX,53:FORT=1TO1000:NEXT:NEXT:P
OKEX,32 :rem 60
365 POKEU+1,59:POKEU-9,240:PRINT"{CLR}":P
OKE214,22:PRINT:PRINT"{ 4 DES}E VISSE
RO A" :rem 35
370 PRINT"{GIU'}{ 2 DES}LUNGO FELICI...":
GOTO460 :rem 126
375 CL=0:X=PEEK(P+1):Y=PEEK(P-1):IFX=62AN
DDR=0THEN350 :rem 227
380 IFX<>39ANDY<>44THENRETURN :rem 133
385 IFX=39THENL=P-21:M=56:GOSUB265
:rem 78
390 IFY=44THENL=P-24:M=56:GOSUB265
:rem 74
395 DR=DR-1 :rem 104
400 POKEU,15:FORX=250TO128STEP-1:POKES1,X
:NEXT:GOSUB325 :rem 13
405 IFPEEK(P+1)=58THENL=P-21:GOSUB275
:rem 228
410 IFPEEK(P-1)=59THENL=P-24:GOSUB275
:rem 230
415 RETURN :rem 122
420 H=INT(RND(1)*414)+7726:BB=INT(RND(1)*
2):X=PEEK(H) :rem 7
425 IFX=37THENL=H:M=41:GOSUB260:RETURN
:rem 190
430 IFX=41THENL=H:M=37:GOSUB260:RETURN
:rem 186
435 IFH<>7731ANDH<>7822ANDH<>7874ANDH<>80
05ANDH<>8106THENRETURN :rem 107
440 IFX<>32ORPEEK(H+1)<>32ORPEEK(H+22)<>3
2ORPEEK(H+23)<>32THENRETURN :rem 162
445 IFBB=1THENL=H:M=37:DR=DR+1:GOSUB265:R
ETURN :rem 188
450 L=H:M=41:DR=DR+1:GOSUB265:RETURN
:rem 3
455 POKEU-9,240:POKEU+1,59:PRINT"{CLR}":P
OKE214,22:PRINT:PRINT"{ 2 DES}I DRAGO
NI VINCONO" :rem 161
460 FORX=1TO23:PRINT:FORT=1TO150:NEXT:NEX
T :rem 36
465 POKEU+1,27:GOTO125 :rem 31
470 CW=(CW+1)AND7:IFCW=1THENCW=2:rem 204
475 POKEZZ+30720,CW:CC=CC+1:IFCC=SLTHEN48
5 :rem 74
480 RETURN :rem 124
485 IZ=INT((ZZ-7680)/22):IP=INT((P-7680)/
22):NZ=INT((IZ+IP)/2)*22 :rem 219
490 NZ=NZ+(P-IP*22+ZZ-IZ*22)/2:CC=1
:rem 22
495 IFPEEK(NZ)=32THENPOKEZZ,32:POKENZ,35:
GOTO510 :rem 66
500 IFPEEK(NZ)=53THENGOSUB330 :rem 44
505 RETURN :rem 122

```

```

510 ZZ=NZ:IFPEEK(ZZ+1)=53ORPEEK(ZZ-1)=53O
RPEEK(ZZ+22)=53ORPEEK(ZZ-22)=53THENG
OSUB330 :rem 54
515 RETURN :rem 123

```

Programma 2

```

1 OPEN1,1,2,"DATI SU NASTRO" :rem 96
2 READX:PRINT#1,X:IFX=-1THEN4 :rem 100
3 GOTO2 :rem 158
4 CLOSE1:PRINT"I DATI SONO STATI":PRINT"R
EGISTRATI SU NASTRO" :rem 157
5 END :rem 12
10 DATA60,66,165,129,153,165,66,60,170,85
,170,85,170,85,170,85,0,56,108,254,15
:rem 69
20 DATA15,31,31,0,0,0,0,1,6,12,152,255,63
,63,31,31,15,31,61,176,176,152,216,220
:rem 40
30 DATA252,248,224,0,0,0,0,128,96,48,25,0
,28,54,127,240,240,248,248,13,13,25,27
:rem 78
40 DATA59,63,31,7,255,252,252,248,248,240
,248,188,0,0,84,124,108,56,56,189
:rem 135
50 DATA21,31,27,31,14,14,14,94,255,255,25
5,255,170,255,255,0,254,254,254,254,17
0
:rem 147
60 DATA254,254,0,0,0,0,0,0,0,255,0,0,0,0
,32,24,6,1,0,0,0,16,8,4,2,1,0,0,8
:rem 194
70 DATA4,4,2,2,1,0,24,126,90,219,24,36,66
,195,0,24,60,90,24,36,0,0,0,0,0
:rem 208
80 DATA24,24,0,0,0,16,0,2,72,2,21,75,21,1
30,16,64,162,72,160,90,224,7,42
:rem 216
90 DATA133,18,5,32,4,0,168,210,168,64,18,
64,8,32,0,40,68,82,5,10,21,10,0,0,0,0
:rem 254
100 DATA1,2,4,16,85,42,21,10,21,10,21,40,
128,144,0,144,72,164,80,160,-1
:rem 153

```

ERRATA CORRIGE

Per un malaugurato errore di impaginazione, nel programma PACMAT versione per C64 pubblicato sul N. 2 della rivista è saltata una parte della linea 430. Diamo qui la linea corretta:

```

430PRINTLEFT$( $$,10)SPC(19)"7777":PRINTS
PC(18)"{SU}8{SIN}{SU}8" :rem 99

```

Per lo stesso motivo, sono state invertite le versioni per VIC 20 e per C64 del programma PATHWAY.

Nel programma MLX la linea 580 contiene un'imperfezione, che comunque non pregiudica il buon funzionamento del programma stesso. A causa di questa imperfezione non si ottiene il numero di checksum indicato sulla rivista. La linea corretta è comunque la seguente:

```

580PRINT" [<+>] "; :rem 79

```

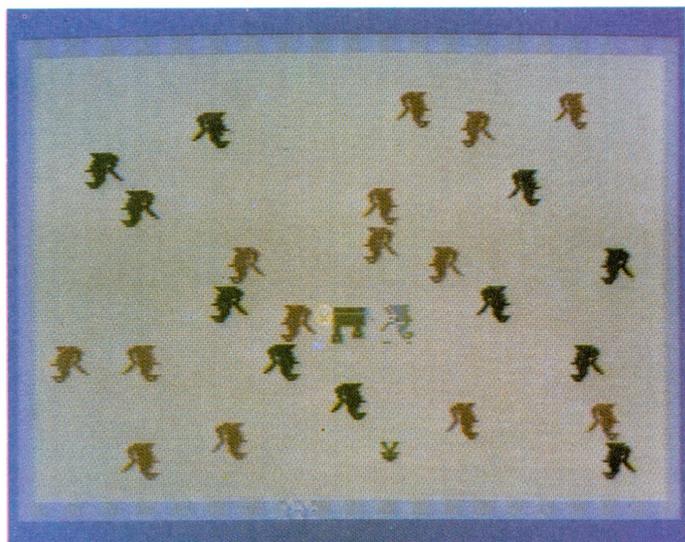
Ci scusiamo con i lettori.

Il Signore dei draghi versione per C64

```

1000 PRINT"{CLR} [<7>]":POKE53280,14:POKE5
3281,6 :rem 137
1010 IFPEEK(14846)=80ANDPEEK(14847)=160TH
EN1070 :rem 143
1020 PRINTTAB(10)"{ 6 GIU'}PER FAVORE ATT
ENDERE":PRINTTAB(9)"MENTRE I DATI VE
NGONO" :rem 135
1025 PRINTTAB(17)"LETTI" :rem 171
1030 FORI=1TO24:READA:NEXT :rem 123
1040 FORD=14616TO14847:READA:POKED,A:NEXT
:POKE52,57:POKE56,57 :rem 216
1050 : :rem 0
1060 : :rem 1
1070 REM RI-INIZIO PROGRAMMA :rem 236
1080 CLR:PRINT"{CLR} [<7>]":DR=24:K=2:P=10
84:S1=54276:S2=54283:GOSUB1260:GOSUB
1490 :rem 237
1090 POKES2-1,13:POKES2+1,8:POKES2+2,0
:rem 193
1100 ZZ=1446:POKEZZ,35:GR(0)=1:GR(1)=12:G
R(2)=11:GR(3)=0:GR(4)=11:GR(5)=12
:rem 147
1110 : :rem 253
1120 V=31-PEEK(56320)AND31:Q=P :rem 173
1130 IFVAND2ANDPEEK(P+40)=32THENP=P+40
:rem 20
1140 IFVAND4ANDPEEK(P-1)=32THENP=P-1
:rem 181
1150 IFVAND1ANDPEEK(P-40)=32THENP=P-40
:rem 25
1160 IFVAND8ANDPEEK(P+1)=32THENP=P+1
:rem 183
1170 POKES2,8:POKEP+54272,GR(GR):GR=GR+1:
IFGR=6THENGR=0 :rem 135
1180 GOSUB1930:IFQ<>PTHENPOKEQ,32:POKEP,5
3:POKES2,65:POKES2-3,3 :rem 186
1190 IFVAND16THENGOSUB1800 :rem 98
1200 IFPEEK(P+1)=37ORPEEK(P+1)=60ORPEEK(P
-1)=42THENGOSUB1640 :rem 99
1210 IFPEEK(P+40)=37ORPEEK(P+40)=60ORPEEK
(P+40)=42THENGOSUB1640 :rem 251
1220 GOSUB2090:GOTO1120 :rem 71
1230 : :rem 0
1240 : :rem 1
1250 REM ISTRUZIONI E LIVELLO DI DIFFICOL
TA' :rem 177
1260 PRINTTAB(3)"*** IL SIGNORE DEI DRAGH
I 64 ***" :rem 136
1270 PRINT"{GIU'} OBIETTIVO: UCCIDERE TUT
TI I DRAGONI ED" :rem 79
1280 PRINT"ENTRARE NEL CASTELLO PER SPOSA
RE LA{ 5 SPAZI}PRINCIPESSA." :rem 227
1290 PRINT"{ 2 GIU'}ELIMINA I DRAGONI MET
TENDOTICI DI FRONTE" :rem 239
1300 PRINT"{SU}E PREMENDO IL PULSANTE DI
SPARO, MA STAI" :rem 195
1305 PRINT"{SU}ALLA LARGA DALLE LORO TEST
E!" :rem 240
1310 PRINT"{ 2 GIU'}PER ULTIMO DEVI ELIMI
NARE IL DRAGONE{ 4 SPAZI}FANTASMA."
:rem 69
1320 PRINT"{ 2 GIU'}{ 2 SPAZI}PREMI: F1 -
- FACILE":PRINTTAB(9)"{GIU'}F3 -- NO
RMALE" :rem 67
1330 PRINTTAB(9)"{GIU'}F5 -- DIFFICILE":P
RINTTAB(9)"{GIU'}F7 -- MOLTO DIFFICI
LE" :rem 188

```



```

1340 SL=15:GETA$:IFA$=""THEN1340:rem 47
1350 IFA$<CHR$(134)ORA$>CHR$(136)THENPRIN
T"{CLR}":RETURN :rem 192
1360 IFA$="{F3}"THENSL=10:PRINT"{CLR}":RE
TURN :rem 233
1370 IFA$="{F5}"THENSL=6:PRINT"{CLR}":RET
URN :rem 192
1380 SL=3:PRINT"{CLR}":RETURN :rem 147
1390 : :rem 7
1400 : :rem 255
1410 REM DISEGNA E CANCELLA I DRAGONI, EC
C. :rem 143
1420 CL=INT(RND(1)*2)*2 :rem 237
1430 CS=L+54272:POKECS,CL:POKECS+1,CL:POK
ECS+40,CL:POKECS+41,CL :rem 121
1440 POKEL,M:POKEL+1,M+1:POKEL+40,M+2:POK
EL+41,M+3:RETURN :rem 243
1450 POKEL,32:POKEL+1,32:POKEL+40,32:POKE
L+41,32:RETURN :rem 61
1460 : :rem 5
1470 : :rem 6
1480 REM INIZIALIZZA IL CAMPO DI GIOCO, I
DRAGONI :rem 123
1490 POKES3265,0:POKE53280,6 :rem 43
1500 POKES3281,0:PRINT"{CLR}":POKE53281,1
5:FORC=14592TO14599:POKEC,0:NEXT:CL=
11 :rem 46
1510 L=1404:M=45:GOSUB1430:POKE1443,49:L=
L-3:M=60:GOSUB1430:F=36 :rem 215
1520 FORX=1024TO1063:POKEX,F:NEXT:FORX=10
64TO1944STEP40:POKEX,F:POKEX+39,F:NE
XT :rem 48
1530 FORX=1984TO2023:POKEX,F:NEXT:FORX=1T
O24:Z=INT(RND(1)*2) :rem 240
1540 READA:M=41:L=1106+A:IFZTHENM=37
:rem 141
1550 GOSUB1420:NEXT:POKE53272,31:POKE1084
,53:POKE1047,53:POKE1048,53:rem 228
1560 DATA1,30,65,82,91,138,203,222,234,30
1,329,345,401,412,424,456,536
:rem 170
1570 DATA550,567,632,690,706,724,734
:rem 61
1580 POKES4296,15:POKE54277,138:POKE54278
,0:POKE53265,27 :rem 165
1590 FORX=49TO52:POKE1443,X:POKE54273,(X-
48)*10:POKES1,8:POKES1,17 :rem 180
1600 FORT=1TO800:NEXT:NEXT:GOSUB1680:RETU
RN :rem 64
1610 : :rem 2
1620 : :rem 3

```

```

1630 REM IL DRAGONE FANTASMA MUORE, VITTO
RIA :rem 180
1640 FORF=1TO2:POKES1,8:POKES1,33:FORX=0T
O40STEP.8:POKE54273,X:NEXT :rem 5
1650 POKES1,8:POKES1,33:FORX=40TO0STEP-.8
:POKE54273,X:NEXT :rem 137
1660 POKEP,F+53:NEXT:POKES1,8 :rem 27
1670 POKE1049-K,36:K=K-1:IFK<0THEN2030
:rem 183
1680 POKEP,32:P=1084:POKEP,53:RETURN
:rem 218
1690 L=P-39:CL=1:M=56:GOSUB1430:GOSUB1860
:L=P-39:GOSUB1450:POKE53272,31
:rem 223
1700 POKE54277,138:POKE54278,0 :rem 151
1710 FORX=52TO49STEP-1:POKE1443,X:POKES1,
8:POKES1,33:POKE54273,(X-48)*10
:rem 70
1720 FORT=1TO800:NEXT:NEXT:POKES1,8
:rem 241
1730 FORX=1441TO1443:POKEX-1,32:POKEX+542
72,1:POKEX,53:FORT=1TO800:NEXT:NEXT
:rem 226
1740 PRINT"{CLR}":POKE53280,7:POKE53281,7
:rem 205
1750 POKE53272,21:PRINT"{CLR} [<3>]"TAB(11
)"{" 22 GIU'}E VISSERO A LUNGO"
:rem 59
1760 PRINTTAB(15)"{GIU'}FELICI...":GOTO20
50 :rem 174
1770 : :rem 9
1780 : :rem 10
1790 REM ELIMINA UN DRAGONE :rem 87
1800 CL=0:X=PEEK(P+1):Y=PEEK(P-1):IFX=62A
NDDR=0THEN1690 :rem 69
1810 IFX<>39ANDY<>44THENRETURN :rem 180
1820 IFX=39THENL=P-39:M=56:GOSUB1430
:rem 173
1830 IFY=44THENL=P-42:M=56:GOSUB1430
:rem 165
1840 DR=DR-1 :rem 148
1850 : :rem 8
1860 POKES1,8:POKES1,129:FORX=20TO0STEP-.
25:POKE54273,X:NEXT:POKES1,8
:rem 64
1870 IFPEEK(P+1)=58THENL=P-39:GOSUB1450
:rem 80
1880 IFPEEK(P-1)=59THENL=P-42:GOSUB1450
:rem 78
1890 RETURN :rem 178
1900 : :rem 4
1910 : :rem 5
1920 REM VOLTA IL DRAGONE / NE CREA UNO N
UOVO :rem 49
1930 H=INT(RND(1)*835)+1106:L=H:BB=INT(RN
D(1)*2):X=PEEK(H) :rem 66
1940 IFX=37THENM=41:GOSUB1420:RETURN
:rem 21
1950 IFX=41THENM=37:GOSUB1420:RETURN
:rem 22
1960 IFRND(1)>.033THENRETURN :rem 214
1970 FORI=-80TO120STEP40:FORJ=-2TO3:IFPEE
K(H+J+I)<>32THENRETURN :rem 228
1980 NEXT:NEXT:M=41:DR=DR+1:IFBB=1THENM=3
7 :rem 144
1990 GOSUB1420:RETURN :rem 52
2000 : :rem 252
2010 : :rem 253
2020 REM I DRAGONI HANNO VINTO :rem 249
2030 PRINT"{CLR}":POKE53280,2:POKE53281,2
:rem 188

```

LOAD

Il signore dei draghi

```

2040 POKE53272,21:PRINT"{CLR} [<3>]"TAB(11
)"{" 24 GIU'}I DRAGONI VINCONO"
:rem 138
2050 FORX=1TO23:PRINT:FORT=1TO150:NEXT:NE
XT:POKE53280,14:POKE53281,6:GOTO1080
:rem 87
2060 : :rem 2
2070 : :rem 3
2080 REM IL MAGO CATTIVO SI SPOSTA
:rem 247
2090 CW=(CW+1)AND15:IFCW=15THENCW=0
:rem 94
2100 CC=CC+1:POKEZZ+54272,CW:IFCC<>SLTHEN
RETURN :rem 241
2110 IZ=INT((ZZ-1024)/40):IP=INT((P-1024)
/40):NZ=INT((IZ+IP)/2)*40 :rem 226
2120 NZ=NZ+(P-IP*40+ZZ-IZ*40)/2:CC=1
:rem 62
2130 IFPEEK(NZ)<>32THENRETURN :rem 97
2140 POKEZZ,32:POKENZ,35:ZZ=NZ:POKEZZ+542
72,CW :rem 197
2150 ZZ=NZ:IFPEEK(ZZ+1)=53ORPEEK(ZZ-1)=53
ORPEEK(ZZ+40)=53THENGOSUB1640
:rem 156
2160 IFZZ>1103ANDPEEK(ZZ-40)=53THENGOSUB1
640 :rem 188
2170 RETURN :rem 170
2180 : :rem 5
2190 : :rem 6
2200 DATA60,66,165,129,153,165,66,60,170,
85,170,85,170,85,170,85 :rem 151
2210 DATA0,56,108,254,15,15,31,31,0,0,0,0
,1,6,12,152 :rem 248
2220 DATA255,63,63,31,31,15,31,61,176,176
,152,216,220,252,248,224 :rem 171
2230 DATA0,0,0,0,128,96,48,25,0,28,54,127
,240,240,248,248 :rem 23
2240 DATA13,13,25,27,59,63,31,7,255,252,2
52,248,248,240,248,188 :rem 95
2250 DATA0,0,84,124,108,56,56,189,21,31,2
7,31,14,14,14,94 :rem 27
2260 DATA255,255,255,255,170,255,255,0,25
4,254,254,254,170,254,254,0:rem 82
2270 DATA0,0,0,0,0,0,255,0,0,0,0,32,24,6,
1,0 :rem 87
2280 DATA0,0,16,8,4,2,1,0,0,8,4,4,2,2,1,0
:rem 213
2290 DATA24,126,90,219,24,36,66,195,0,24,
60,90,24,36,0,0 :rem 237
2300 DATA0,0,0,24,24,0,0,0,16,0,2,72,2,21
,75,21 :rem 245
2310 DATA130,16,64,162,72,160,90,224,7,42
,133,18,5,32,4,0, :rem 58
2320 DATA168,210,168,64,18,64,8,32,0,40,6
8,82,5,10,21,10 :rem 230
2330 DATA0,0,0,0,1,2,4,16,85,42,21,10,21,
10,21,40 :rem 86
2340 DATA128,144,0,144,72,164,80,160
:rem 23

```



SPECTRUM

EDUCATIVI/DIDATTICI

GRAFICA PER TUTTI

Il vostro Spectrum diventa un potente ma semplice strumento per disegnare, e il vostro schermo una tela, sulla quale dipingere con i colori dell'arcobaleno. Un programma italiano, pensato soprattutto per la didattica, facile da usare e adatto anche per i più piccoli, ma che può sfruttare istruzioni potenti che permettono, ad esempio, di colorare una figura solo definendo un punto all'interno della stessa.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0100-01 L. 25.000

MANUALE DI GEOMETRIA PIANA

Il programma consente la consultazione e l'applicazione pratica di numerose regole di geometria piana, tra cui: calcolo di aree, perimetri, settori, ecc. Valido aiuto agli studenti e professionisti per fare rapidamente i calcoli.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0100-02 L. 25.000

MANUALE DI GEOMETRIA SOLIDA

Il programma consente la consultazione e l'applicazione pratica di numerose regole di geometria solida, tra cui: calcolo di volumi, superfici, sezioni, ecc. Aiuto incontestabile per studenti, professionisti e chiunque abbia da affrontare questi problemi e i relativi calcoli.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0100-03 L. 25.000

TRIGONOMETRIA

Eccezionale TRILOGIA che trasporta l'utente nel mondo incantato della TRIGONOMETRIA. Valido aiuto per lo studente che deve affrontare questa materia per la prima volta. Manuale pratico per il professionista che deve risolvere problemi trigonometrici. Il programma offre il vantaggio non solo di risolvere i triangoli ma anche di visualizzarli ridotti in scala. La TRILOGIA viene fornita con un utile volumetto, che riproduce tutte le principali vedute, al fine di migliorare la consultazione del programma.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0100-04 L. 25.000

GIOCHI

MOAICO

Il programma moaico è un gioco che vi permette di scoprire pezzo per pezzo, un disegno precedentemente realizzato con lo Spectrum. I disegni possono essere già presenti sulla cassetta, oppure realizzati dall'utente, con un apposito programma contenuto nella cassetta stessa. Le numerosissime caselline che compongono il disegno, rendono appassionante il gioco, la possibilità di creare sempre nuovi disegni, lo rende elettrizzante.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0101-01 L. 20.000

BATTAGLIA NAVALE

Evitiamo lo spreco di carta e le laboriose sistemazioni delle navi. Adesso è possibile giocare alla Battaglia Navale con lo Spectrum. Il calcolatore segnala i tiri effettuati e i centri ottenuti. Un giocatore per volta definisce la posizione delle proprie navi. Il calcolatore memorizzerà le posizioni delle navi e per ogni tiro darà l'adeguato messaggio. Quando un giocatore affonda tutte le navi dell'avversario compare la scritta "VITTORIA" e vengono visualizzate le navi superstiti del vincitore.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0101-02 L. 20.000

PUZZLE MUSICALE

Programma che, oltre ad essere un gioco, possiede ottime caratteristiche didattiche offrendo la possibilità di imparare e riprodurre i brani proposti dal computer.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0101-03 L. 20.000

SUPER EG

Sei atterrato su Marte e hai scoperto un labirinto in cui gli antichi abitanti hanno lasciato un favoloso tesoro custodito da molti alieni. Con l'aiuto di una mappa elettronica devi percorrere le varie stanze difendendoti dai mostri e devi raccogliere le sette chiavi che aprono il cofano del tesoro. Il primo gioco interamente ideato e programmato in Italia con qualità grafica e controllo di movimento paragonabile ai migliori giochi arcade in circolazione.
J/0101-04 L. 20.000

APPLICATIVI/GESTIONALE

SPECTRUM WRITER

MICRODRIVE COMPATIBILE
SPECTRUM WRITER è il programma professionale di elaborazione e stampa delle parole e dei testi - Word Processor - creato specificatamente per il computer Spectrum 48 K. SPECTRUM WRITER consente di scrivere e comporre qualsiasi tipo di testo. Sono immediatamente collegabili la ZX PRINTER, la SEIKOSHA 505 e la ALPHACOM 32. Con l'apposita interfaccia è possibile collegare qualsiasi tipo di stampante parallela.
Supporto: cassetta
Configurazione richiesta: 48 K RAM, microdrive opzionale
J/0102-01 L. 40.000

MASTER FILE - MICRODRIVE COMPATIBILE

Senza dubbio il più potente data base e sistema di archiviazione di files oggi disponibile. Interamente scritto in codice macchina per il compatimento e la velocità, offre 32 K di memoria - max - per i dati in ogni file - 26 campi per record - 128 caratteri per campo. I files possono essere caricati o salvati indipendentemente dal programma e memorizzati su cassette o cartridge Microdrive.
Supporto: cassetta
Configurazione richiesta: 48 K RAM, microdrive opzionale
J/0102-02 L. 40.000

TECNICO/SCIENTIFICI

ING. 1/ CALCOLO AD ELEMENTI FINITI

Questo programma, adoperando il metodo ad elementi finiti triangolari, permette di analizzare in maniera precisa e veloce pannelli piani di qualsiasi forma e di qualsiasi caratteristiche elastiche. La struttura può essere discretizzata con più di ottanta elementi permettendo così la risoluzione di problemi anche iperstatici con una geometria complessa. La possibilità di rappresentare graficamente la struttura deformata, sovrapposta a quella non deformata, permette una facile interpretazione dei risultati e del comportamento della stessa.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-01 L. 30.000

TOPOGRAFIA

Programma veramente completo e utile a tutti gli studi professionali, questo Package permette il calcolo dell'area di una figura piana in molteplici modi, a seconda dei dati disponibili. Strutturato a sottoprogrammi, indipendenti fra loro, il programma consente il calcolo di aree con il metodo del camminamento, fornendo le coordinate cartesiane o polari dei vertici, o in svariati altri modi. La rappresentazione grafica della figura di cui si sta calcolando l'area, e la possibilità di memorizzare e sommare fra loro aree di figure diverse completano il programma.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-02 L. 30.000

CALCOLO TRAVI IPE

Il programma consente il calcolo di travi IPE, caricate uniformemente e semplicemente appoggiate agli estremi. Si può effettuare il calcolo di una sola trave, o di più travi affiancate.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-03 L. 25.000

ENERGIA SOLARE

Finalmente un programma che consente il calcolo di un impianto solare in brevissimo tempo, che non si limita ad una semplice analisi, ma anche ad una valutazione economica. Si è tanto parlato di energie alternative, ma pochi ne hanno parlato in modo completo anche dal punto di vista economico.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-04 L. 30.000

ALGEBRA MATRICIALE

Raccolta di otto programmi che permettono di risolvere le operazioni dell'algebra matriciale:
1 - Matrice inversa
2 - Determinante
3 - Prodotto
4 - Somma
5 - Sistemi di equazioni - Metodo di GAUSS
6 - Sistemi di equazioni - Metodo iterativo di GAUSS/SEIDEL
7 - Autovalori complessi
8 - Decomposizione ortogonale
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-05 L. 30.000

STUDIO DI FUNZIONI

E' un programma per disegnare in alta risoluzione fino a 8 grafici di funzioni diverse. Si possono trovare MASSIMI, MINIMI, INTERSEZIONI, ZERI, ecc. con precisione a piacere. E' possibile determinare anche il campo di esistenza e i limiti della funzione e si può ingrandire a tutto schermo qualsiasi piccolo particolare del grafico. E' completamente protetto da qualsiasi tipo di errore.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0104-06 L. 30.000

HOME/DOMESTICO

TOTIP

Un programma per giocare la schedina Totip; una colonna per volta oppure dei sistemi con triple e doppie. All'inizio del programma, l'utente propone le probabilità di uscita in ogni segno, per ogni corso, secondo il suo giudizio. Il programma emette la schedina in base a quelle probabilità e ad un algoritmo che stabilisce le cosiddette "sorprese". Quindi, la schedina che ne deriva è ragionata, e allo stesso tempo casuale.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0105-01 L. 20.000

ASTROLOGIA

Oroscopo accuratissimo e personalissimo di almeno 2000 parole: viene calcolato il vostro personale oroscopo, completo di tutti gli aspetti interplanetari e le configurazioni relative allo zodiaco, le loro longitudini esatte a meno di 6 minuti d'arco e la loro interpretazione. Sistemi usati: zodiaco tropicale e sistema di Placidus per la divisione delle Case.
Supporto: cassetta
Configurazione richiesta: 48 K RAM
J/0105-02 L. 25.000

CBM 64

GIOCHI

GARDEN WARS

Una fuga su 8 piani diversi, inseguito da mostri orrendi per sfuggire da un incantesimo, aiutato solo dalla propria abilità e da alcuni gnomi. Ecco il compito del nostro eroe. Un gioco, che con solo due livelli di difficoltà, selezionabili dall'utente, riesce a soddisfare le esigenze dei neofiti, e dei più esperti giocatori di videogames. La grafica del C64 viene in questo programma sfruttata al meglio rendendolo accattivante.
Supporto: cassetta
Configurazione richiesta: standard
J/0111-01 L. 20.000

VIC 20

GIOCHI

CHESS WARS

Finalmente è possibile giocare a scacchi sul VIC20 inespanso. Questo programma permette di giocare con il calcolatore selezionando fino a 9 livelli di difficoltà. Il programma comprende tutte le mosse speciali come, promozione di pedone a regina, en-passant, arrocco stabilite dal regolamento, ed alcune mosse di apertura standard: la partita spagnola, la difesa siciliana, la difesa francese e la partita di Regina ortodossa. Un modo divertente per trascorrere le giornate in compagnia del vostro VIC20.
Supporto: cassetta
Configurazione richiesta: RAM standard
J/0121-01 L. 20.000

Cedola di commissione da inviare a:
JCE - Via dei Lavoratori, 124 - 20092 Cinisello B. - MI
Inviatemi i seguenti programmi:

Cod. progr.	Q.tà	Cod. progr.	Q.tà	Cod. progr.	Q.tà
Cod. progr.	Q.tà	Cod. progr.	Q.tà	Cod. progr.	Q.tà
Cod. progr.	Q.tà	Cod. progr.	Q.tà	Cod. progr.	Q.tà

Desidero ricevere il materiale indicato nella tabella, a mezzo pacco postale contro assegno, al seguente indirizzo:

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA
Partita I.V.A.

PAGAMENTO:
 Anticipato, mediante assegno circolare o vaglia postale per l'importo totale dell'ordinazione
 Contro assegno, al postino l'importo totale
AGGIUNGERE L. 2.500 per contributo fisso spedizione. I prezzi sono comprensivi di I.V.A.



Via dei Lavoratori, 124
20092 Cinisello Balsamo - MI

SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE

CBM 64

EDUCATIVI

TURTLE GRAPHICS II

Un modo eccitante e divertente per introdurre il principiante alla scoperta dei concetti base di programmazione. Semplice da utilizzare, questo nuovo linguaggio didattico comprende le migliori caratteristiche del LOGO e PILOT. Aggiunge più di 30 comandi differenti per il sonoro, colore, grafica, movimento ecc.
Supporto: cartridge
Configurazione richiesta: standard
H/0310-01 L. 74.000

TURTLE TOYLAND JR

Guidando una tartaruga lungo lo schermo con il joystick è possibile creare, muovere ed inserire nuove figure e caratteri. Un metodo nuovo per apprendere i concetti del computer e della programmazione giocando con i colori, gli effetti musicali e la grafica in movimento sullo schermo.
Supporto: cassetta e disk
Configurazione richiesta: standard, joystick, registratore o floppy disk 1541.
H/0310-02 L. 74.000

PAINT BRUSH

Un generatore grafico studiato per creare disegni sullo schermo, consigliato a tutte le età - dai 6 anni in poi. Utilizzando il joystick per controllare i movimenti del cursore e la tastiera per selezionare i colori e la grafica, le possibilità di questo package diventano praticamente illimitate. A corredo viene inoltre fornita, una cassetta dimostrativa contenente numerosi esempi grafici e disegni.
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0310-03 L. 45.000

GIOCHI

RETROBALL

L'emozionante e divertente gioco dell'Hockey su ghiaccio riproposto ora sul computer.
Possibilità di giocare fra 2 giocatori o contro il computer.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
H/0311-01 L. 24.000

MR. TNT

Mr. TNT è la versione da casa del famoso gioco arcade giapponese "DYNAMITE". Conduci il candelotto di TNT attraverso il circuito di piste ardenti muovendo molto velocemente, poiché una piccola indecisione li farebbe accendere distruggendoti.
Venti livelli di difficoltà crescente ti attendono!
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0311-02 L. 70.000

GHOST MANOR/SPIKE'S PEAK

Due splendide avventure grafiche al prezzo di una. In Ghost manor il giocatore deve affrontare spettri e scheletri, mostri mummie e formiche mortali per salvare l'amico dal Conte Dracula. In Spike's Peak una straordinaria sequenza di catastrofi naturali e pericoli mortali.
Supporto: diskette
Configurazione richiesta: floppy disk 1541
H/0311-03 L. 60.000

MINNESOTA FATS' POOL CHALLENGE

Un simulatore di biliardo con tutte le caratteristiche reali di un vero biliardo.
Sono previsti 1 o 2 giocatori con 6 varianti di gioco.
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0311-04 L. 70.000

THE PIT

In questo famoso gioco arcade bisogna condurre l'avventuriero nel tunnel sotterraneo di un pianeta alieno alla ricerca del favoloso forziere che però contiene PIT, una trappola mortale dalla quale nessuno è ancora riuscito a sfuggire.
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0311-05 L. 70.000

ROOTIN TOOTIN

Un popolare gioco arcade giapponese con schermi e concetti completamente inediti nel campo dei giochi/labirinto. Ti sposti lungo i righe del pentagramma con un trombone colpendo le note che incontri davanti a te.
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0311-06 L. 80.000

HES GAMES 84

Al meglio della condizione fisica ti presenti alla competizione di queste 6 discipline olimpiche offerte da HES GAMES: tuffi, sollevamento pesi, 100 metri piani, salto in lungo, tiro con l'arco, corsa ad ostacoli.
Supporto: diskette
Configurazione richiesta: floppy disk 1541, joystick
H/0311-07 L. 90.000

SYNTHESOUND 64

Trasforma il tuo C64 in un incredibile sintetizzatore musicale e generatore di effetti speciali.
La tastiera del computer diventa una vera e propria tastiera musicale con una parte solista ed una parte per l'accompagnamento.
Supporto: diskette
Configurazione richiesta: standard
H/0311-08 L. 45.000

COCO

Eccellente computer games di tipo educativo, che insegna le tecniche fondamentali del computer, il linguaggio basic, come risolvere i problemi scomponendo ogni programma in semplici pezzi e simulando l'effetto di ogni istruzione.
Supporto: cassetta
Configurazione richiesta: standard
Istruzioni in italiano.
H/0311-09 L. 35.000

BENJI'S SPACE RESCUE

Assieme al cane Benji* stella di molti sceneggiati televisivi americani, intraprenderà un viaggio attorno al sistema solare, alla ricerca di alcuni scienziati rapiti. Scoprirai le meraviglie del cosmo, e combatterai contro alieni, insieme al fedele Benji. Dovrai mettercela veramente tutta per scoprire dove sono tenuti prigionieri gli scienziati. Ce la farai?
Supporto: diskette
Configurazione richiesta: floppy disk 1541
H/0311-10 L. 24.000

APPLICATIVO GESTIONALE

HES WRITER 64

Con questo programma avanzato di word processing è possibile scrivere molto facilmente note, lettere ed ogni tipo di manoscritto.
Supporto: cartridge
Configurazione richiesta: registratore o floppy disk 1541 per il salvataggio dei testi.
Stampante Commodore o compatibile.
H/0312-01 L. 100.000

OMNI WRITER/OMNI SPELL

Un potentissimo programma di word processing estremamente facile da utilizzare. Puoi creare qualsiasi tipo di testo, lettere commerciali, articoli di giornali e stampare tutte le copie desiderate. OMNI WRITER è semplice da usare come una macchina da scrivere ma aggiunge la flessibilità del computer. Puoi cambiare istantaneamente il contenuto dello schermo e vedere che cosa stamperà.
Supporto: diskette
Configurazione richiesta: floppy disk 1541
Stampante Commodore o compatibile.
H/0312-02 L. 160.000

UTILITA'

HESMON 64

HESMON è un monitor linguaggio macchina del 6502/6510 con mini assembler e con l'aggiunta di 30 comandi di utilità. Un potente accessorio con molte caratteristiche non reperibili sugli altri monitor ed indispensabile per tutti coloro che vogliono programmare in linguaggio assembler.
Supporto: cartridge
Configurazione richiesta: standard
H/0313-01 L. 90.000

64 FORTH

Linguaggio interattivo che unisce la facilità del Basic alla velocità ed alla efficienza nell'uso della memoria dell'assembler.
Supporto: cartridge
Configurazione richiesta: standard
H/0313-02 L. 140.000

6502/6510 PROFESSIONAL DEVELOPMENT SYSTEM

Un completo package per il linguaggio assembler, ideale per i principianti e indispensabile per i programmatori esperti che racchiude 2 programmi: HESBAL e HESDIT.
Supporto: cassetta
Configurazione richiesta: standard
H/0313-03 L. 70.000

VIC 20

EDUCATIVI

TURTLE GRAPHICS

Un modo eccitante e divertente per introdurre il principiante alla scoperta dei concetti base di programmazione. Semplice da utilizzare, questo nuovo linguaggio didattico comprende le migliori caratteristiche del LOGO e PILOT.
Supporto: cartridge
Configurazione richiesta: espansione RAM opzionale
H/0320-01 L. 90.000

GIOCHI

TORG

TORG è un'avventura all'interno di un intricato labirinto. Intrappolato dentro una griglia devi assolutamente cercare la via d'uscita distruggendo le varie ondate di mostri che ti attaccano per annientarti.
Supporto: cassetta
Configurazione richiesta: RAM standard, joystick
H/0321-01 L. 15.000



RAID ON ISRAM

Una missione di ricerca e distruzione controllata interamente dal joystick. Con l'aiuto dei lanciarazzi, piloti il tuo caccia in una fantastica avventura ai limiti del reale.
Supporto: cassetta
Configurazione richiesta: RAM standard, joystick
H/0321-02 L. 15.000

COCO 2

Un fantastico computer games di tipo educativo, che ti permetterà di creare da solo i tuoi giochi.
Supporto: cassetta e disk
Configurazione richiesta: 16K RAM, registratore o floppy disk 1541
H/0321-03 L. 30.000

AGGRESSOR

Come un valoroso pilota devi proteggere i depositi di benzina di Stellarium dagli attacchi delle navicelle di Zaurien.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
H/0321-04 L. 30.000

GRID RUNNER

Accanitevi contro il nemico droido che si ammassa su "Grid" la stazione nucleare nell'orbita solare terrestre.
Supporto: cartridge
Configurazione richiesta: standard, joystick
H/0321-05 L. 50.000

PROTECTOR

Le orde della città di Fraxullan stanno attaccando la tua città e catturando gli abitanti. Tu devi aiutare la popolazione, trasportandola nella Città della Nuova Speranza, sull'altro lato del vulcano; ma i sopravvissuti non sono ancora salvi.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
H/0321-06 L. 30.000

PREDATOR

Fiero come un'aquila, voli nel cielo, difendendo il tuo spazio dagli attacchi di nemici pennuti. Mano a mano che sconfiggi i tuoi avversari, avanzi lungo la scala di 99 livelli di difficoltà.
Supporto: cartridge
Configurazione richiesta: standard, joystick
Istruzioni in italiano.
H/0321-07 L. 30.000

APPLICATIVO GESTIONALE

HES WRITER

Con questo programma avanzato di word processing è possibile scrivere molto facilmente note, lettere ed ogni tipo di manoscritto. Alcune delle caratteristiche sono: centratura del testo, numerazione delle pagine, cancellazione e spostamento di una o più linee di testo, ricerca veloce di stringhe.
Supporto: cartridge
Configurazione richiesta: espansione RAM opzionale; registratore o floppy disk 1541 per il salvataggio dei testi; stampante Commodore compatibile.
Istruzioni in italiano.
H/0322-01 L. 50.000

UTILITA'

6502 PROFESSIONAL DEVELOPMENT SYSTEM

Un completo package per il linguaggio assembler, ideale per i principianti e indispensabile per i programmatori esperti che racchiude 2 programmi: HESBAL e HESDIT. HESBAL è un assembler che usa i mnemonici standard MOS in 1 o 2 passi
HESDIT: un potente FULL-SCREEN TEXT EDITOR indispensabile per creare o gestire numerosi files,
Supporto: cassetta
Configurazione richiesta: standard
H/0323-01 L. 50.000

VIC FORTH

Linguaggio interattivo che unisce la facilità del basic alla velocità ed alla efficienza nell'uso della memoria dell'assembler.
Supporto: cartridge
Configurazione richiesta: espansione RAM opzionale
Manuale d'istruzioni in italiano
H/0323-02 L. 50.000

HESMON

HESMON è un monitor linguaggio macchina del 6502 con un mini assembler e con l'aggiunta di 30 comandi di utilità.
Supporto: cartridge
Configurazione richiesta: standard
Manuale d'istruzione in italiano
H/0323-03 L. 50.000



Lista variabili

di E. A. Cottrell
trad. ed adatt. di U. Barzagli

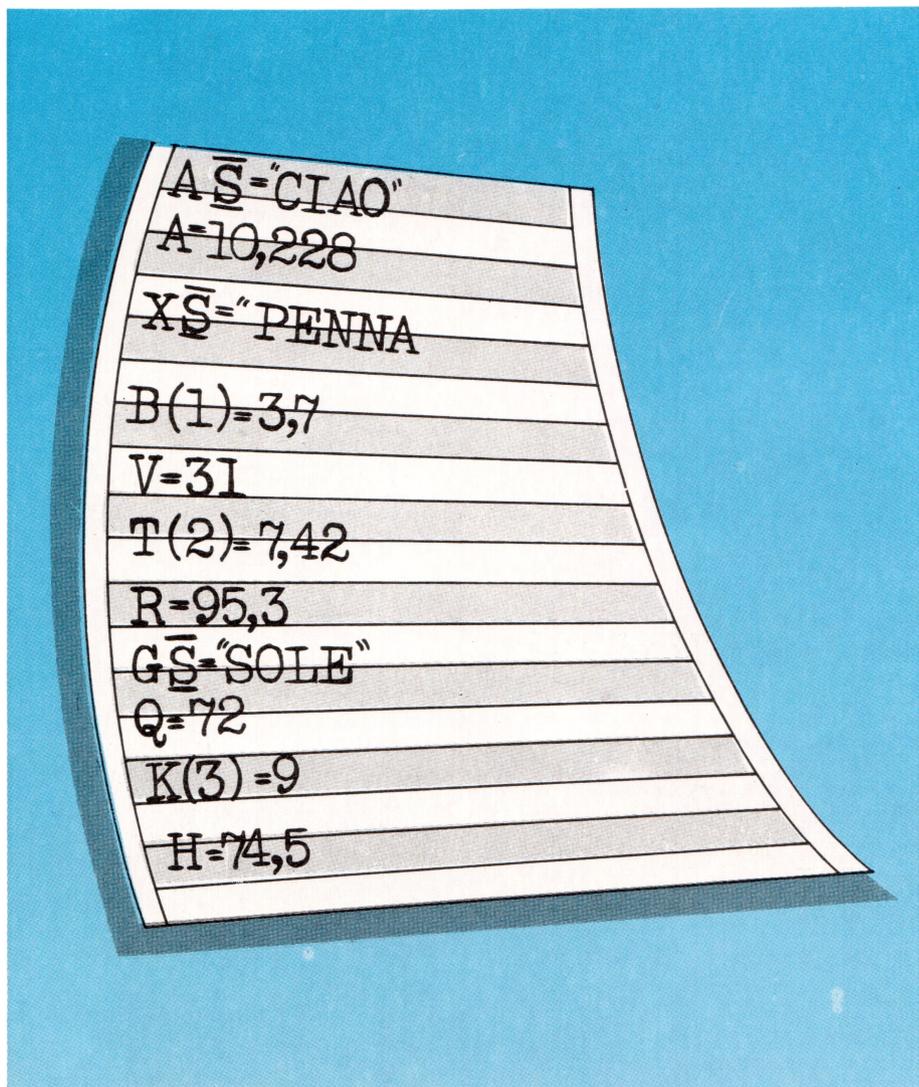
Questo programma di utilità pratica lista tutte le variabili contenute nel vostro programma, incluso il loro tipo (semplice, di stringa, a più dimensioni). È un utile supporto, soprattutto per il controllo di programmi molto lunghi o per scrivere la documentazione dei vostri programmi. È adatto sia al VIC che al C64.

Ci sono due tipi di variabili, semplici ed a più dimensioni, e tre categorie per ogni tipo, numeriche a virgola mobile, numeriche intere e alfanumeriche. Tutte queste variabili vengono immagazzinate nella memoria del VIC e del C64 immediatamente dopo il programma BASIC.

Le variabili semplici vengono memorizzate al di sotto delle variabili a più dimensioni, a partire dalla locazione individuata dal puntatore contenuto nelle locazioni di memoria 45 e 46. Ognuna di queste variabili semplici occupa sette byte di memoria. I primi due byte contengono i primi due caratteri (in ASCII) del nome della variabile, con una codifica che ne indica il tipo. Questa codifica è semplicemente ottenuta aggiungendo 128 al codice ASCII di entrambe i caratteri se si tratta di una variabile intera, e sommando 128 al solo secondo carattere, se si tratta di una variabile alfanumerica. Nessuna codifica contraddistingue le variabili a virgola mobile. I byte rimanenti contengono il valore attuale della variabile. Nel caso di una variabile alfanumerica, i byte rimanenti contengono la lunghezza della stringa e la locazione all'inizio della memoria che contiene il primo carattere della stringa.

Le variabili a più dimensioni o matrici sono piuttosto diverse, in quanto le dimensioni della variabile sono determinate dal numero di elementi della matrice. Le informazioni che devono essere memorizzate per una variabile a più dimensioni comprendono il nome della variabile, che viene codificato nello stesso modo di una variabile semplice, un puntatore alla locazione di memoria che contiene la variabile successiva, il numero delle dimensioni della matrice, ed il numero degli elementi contenuti nella matrice.

Inoltre, deve essere memorizzato il va-



lore di ogni elemento nel caso di matrici numeriche, o il puntatore alla stringa e la sua lunghezza nel caso di matrici alfanumeriche. Come potete vedere le variabili a più dimensioni si possono mangiare gran parte dello spazio in memoria. È conveniente utilizzare il minor numero possibile di elementi nelle matrici. Se non specificate le dimensioni di una matrice, il calcolatore la fisserà a dieci elementi. Se ne avete bisogno meno di dieci, risparmierete un minimo di cinque byte

per elemento se stabilirete le dimensioni della matrice con una istruzione DIM (=dimensionamento). Anche se una variabile semplice intera occupa nel complesso lo stesso spazio in memoria di una variabile a virgola mobile, si possono risparmiare tre byte per elemento se utilizzate variabili intere al posto di quelle in virgola mobile nelle matrici.

Il caricamento del programma
"Lista Variabili" è un programma in

linguaggio macchina che viene caricato tramite delle istruzioni POKE da un programma BASIC, la qual cosa elimina la necessità di disporre di un assembler. Il linguaggio macchina è automaticamente caricato alla fine della memoria e protetto dai vostri programmi BASIC. Prima di mandare in esecuzione con il RUN questo programma, assicuratevi di averne registrata una copia poiché esso autodistrugge dopo essere stato eseguito. Dopo che la parte in linguaggio macchina sia stata caricata, il programma caricatore vi segnalerà la locazione di memoria da attivare con una istruzione SYS quando volete la lista delle vostre variabili. Per esempio, con una espansione di memoria da 16K inserita nel vostro VIC, dovrete battere SYS24320 per ottenere la lista delle variabili. Il programma elencherà quindi le variabili semplici nell'ordine in cui appaiono nel programma stesso, con indicazioni circa il loro tipo. Successivamente vengono elencate le variabili a più dimensioni con indicatori di tipo opportuni.

"Lista Variabili" è particolarmente utile quando scrivete programmi con molte variabili e vi trovate nella necessità di inventare nuovi nomi per le variabili stesse. È anche un valido strumento per la documentazione dei programmi una volta terminati.

Le variabili vengono elencate in senso orizzontale sullo schermo per evitare che scorrano oltre ad esso (scrolling).

Se avete una stampante, le modifiche seguenti possono essere apportate al programma originale per darvi un elenco di più facile lettura.

```
160 IF PA { } 33632 THEN PRINT "ERRORE NEI DATI": END
260 DATA 32,210,255,169,13,32,210
420 DATA 41,32,210,255,169,13,32,
```

Per inviare l'elenco alla vostra stampante, limitatevi ad aprire un file per la stampante stessa:

```
OPEN1,4:CMD1:SYS XXXXX
```

Il programma BASIC di cui volete ottenere l'elenco delle variabili deve essere mandato in esecuzione con il RUN prima che voi attivate il programma "Lista variabili" con una SYS. Questo perché le variabili non sono inserite in memoria finché un programma non viene mandato in esecuzione.

Puntatori ad indirizzi di memoria

Più volte incontrerete riferimenti a "puntatori" all'interno della memoria di un calcolatore. Essi sono dei numeri di due byte di lunghezza, generalmente posizionati nelle prime 256 celle di memoria del computer, che contengono degli importanti indirizzi.

La situazione muta nel corso dell'esecuzione di un programma o nel corso della sua scrittura. Ad esempio, se aggiungete una riga ad un programma BASIC, voi avete aumentato lo spazio

totale che il programma occupa nella memoria ad accesso casuale (RAM). Ovviamente, quando vi accingete a registrare il programma, il calcolatore deve sapere dove il programma BASIC termina. Quindi, registra la "fine attuale del programma BASIC" tramite un puntatore. Questo puntatore è posizionato (sia nel VIC che nel C64) agli indirizzi di memoria 45 e 46. Il numero contenuto nella cella 46 viene moltiplicato per 256 e sommato a quello contenuto nella cella 45. Per sapere a quale indirizzo della memoria RAM termina il vostro attuale programma BASIC, potete battere: andare a capo e lasciare spazi sopra e sotto ?PEEK (45) + PEEK (46) * 256.

C'è anche un certo numero di altri puntatori, comprendente "limite della memoria", "inizio delle matrici", "registro stringhe alfanumeriche" e "inizio del BASIC". Le locazioni di questi puntatori sono elencate nelle "mappe di memoria" di ogni calcolatore. Sono anche spesso disponibili presso i gruppi di utenti. Ci sono alcune cose interessanti che potete fare manipolando questi puntatori con delle istruzioni POKE. Ad esempio, potete ingannare il calcolatore riservando spazio in memoria in modo che due programmi BASIC indipendenti possano essere eseguiti simultaneamente. Comunque, i puntatori contengono informazioni essenziali per il calcolatore, ed il loro valore può essere reso accessibile utilizzando la formula suddetta.

Lista variabili versione per VIC 20 e C64

```
120 ME=PEEK(55)+256*PEEK(56) :rem 21
130 VS=ME-256:PA=0 :rem 14
140 POKE56,PEEK(56)-1 :rem 154
150 FORI=VSTOVS+240:READA:POKEI,A:PA=PA+A :rem 237
:NEXT :rem 237
160 IFPA<>33670THENPRINT"ERRORE NEI DATI" :rem 18
:END :rem 18
170 PRINT"SYS"VS"PER ATTIVARE":NEW :rem 192
:rem 192
180 DATA165,45,197,47,240,93,133:rem 101
190 DATA253,165,46,133,254,160,0:rem 85
200 DATA169,0,141,61,3,177,253 :rem 237
210 DATA41,128,208,60,177,253,41:rem 83
220 DATA127,32,210,255,200,173,61 :rem 120
:rem 120
230 DATA3,201,0,208,6,177,253 :rem 182
240 DATA41,128,208,46,177,253,41:rem 90
250 DATA127,32,210,255,173,61,3 :rem 28
260 DATA32,210,255,169,32,32,210:rem 75
270 DATA255,152,24,105,6,144,5 :rem 239
280 DATA164,254,200,132,254,168,101 :rem 230
:rem 230
290 DATA253,197,47,240,17,208,186 :rem 154
:rem 154
300 DATA96,169,37,141,61,3,208 :rem 248
310 DATA189,169,36,141,61,3,208 :rem 43
```

```
320 DATA203,165,49,197,47,240,114 :rem 141
:rem 141
330 DATA165,47,133,253,165,48,133 :rem 143
:rem 143
340 DATA254,160,0,169,0,141,61 :rem 232
350 DATA3,177,253,240,216,41,128:rem 85
360 DATA208,77,177,253,41,127,32:rem 96
370 DATA210,255,200,173,61,3,201:rem 69
380 DATA0,208,6,177,253,41,128 :rem 246
390 DATA208,63,177,253,41,127,32:rem 94
400 DATA210,255,173,61,3,32,210 :rem 18
410 DATA255,169,40,32,210,255,169 :rem 139
:rem 139
420 DATA41,32,210,255,169,32,32 :rem 27
430 DATA210,255,200,177,253,24,101 :rem 171
:rem 171
440 DATA253,197,49,240,39,177,253 :rem 157
:rem 157
450 DATA24,101,253,170,200,177,253 :rem 176
:rem 176
460 DATA101,254,133,254,134,253,208 :rem 231
:rem 231
470 DATA165,96,169,37,141,61,3 :rem 2
480 DATA208,172,169,36,141,61,3 :rem 43
490 DATA208,186,165,48,197,50,208 :rem 160
:rem 160
500 DATA136,96,200,234,177,253,101 :rem 182
:rem 182
510 DATA254,197,50,240,224,16,222 :rem 132
:rem 132
520 DATA136,208,202 :rem 209
:rem 209
```



Cylon Zap

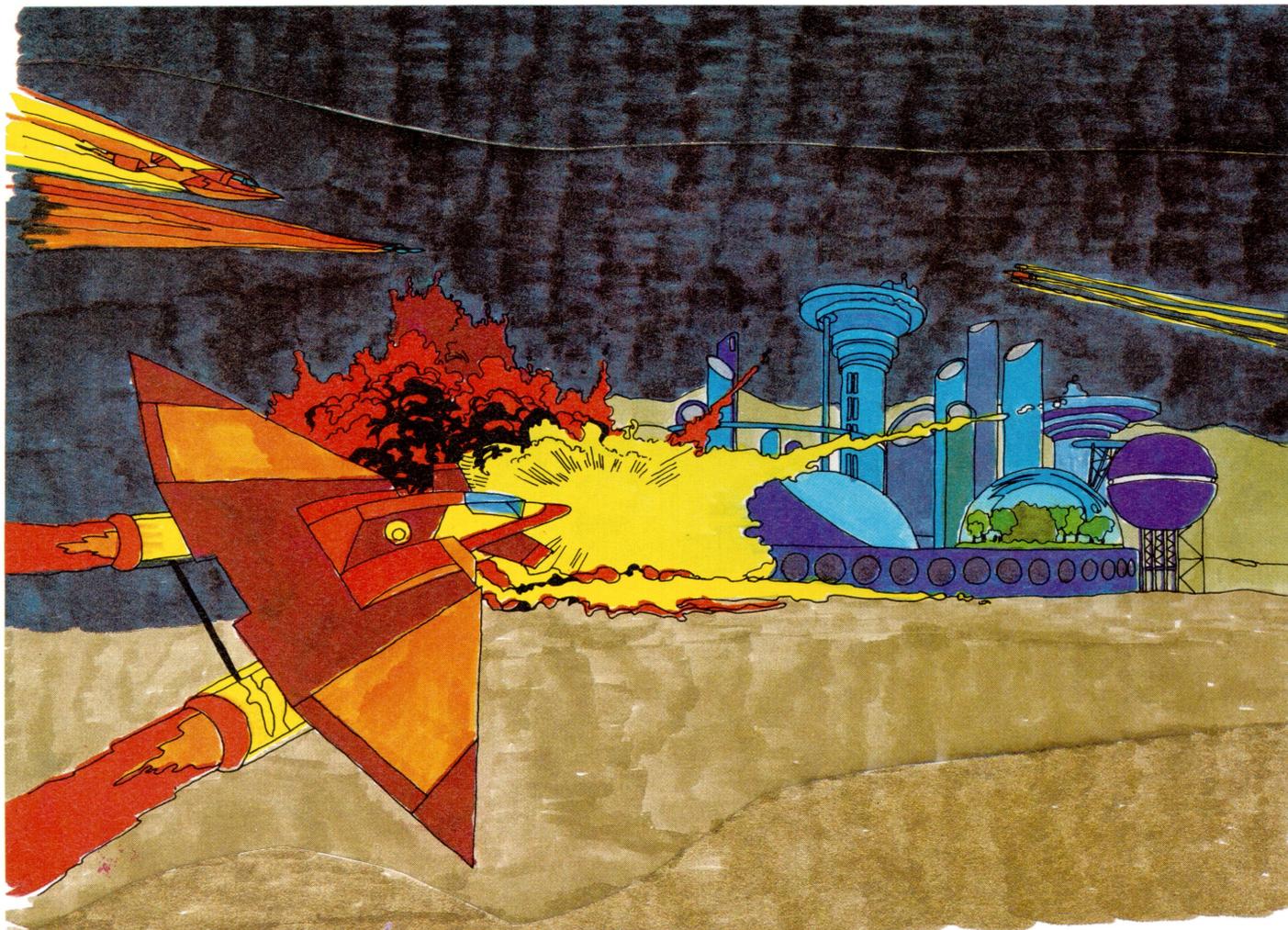
di **M. Dudley**

trad. ed adatt. di **U. G. Barzaghi**

Cylon Zap è un gioco simile a quelli diffusi nelle sale-giochi (in inglese "arcade") per il VIC 20 fornito di espansione di memoria da 8K byte, e per il Commodore 64. Una stazione spaziale al centro dello scher-

mo, che voi dovete difendere a qualunque costo, è continuamente attaccata dai caccia di Cylon. Dovete abatterli prima che si buttino (tipo kamikaze) sulla stazione spaziale.

Per difendervi dai Cyloni, avete due armi a disposizione. La prima è rappresentata dal joystick, che può essere spostato su, giù, a destra ed a sinistra per azionare i laser in ciascuna di queste quattro direzioni. La seconda è co-



stituita dal pulsante di sparo del joystick, che innesca una bomba "F" (l'iniziale sta per "furba"), che libera immediatamente lo schermo da ogni attaccante. Le bombe "F" devono essere usate con parsimonia, poiché ce ne sono solo tre disponibili all'inizio del gioco.

Il punteggio ed il numero di bombe rimanenti vengono costantemente aggiornati nell'angolo in alto a sinistra dello schermo. Quando il punteggio raggiunto è 30, gli assalitori sui fianchi iniziano ad aumentare la loro velocità. Quando è 50, anche gli attaccanti dall'alto e dal basso aumentano la loro velocità. Se il vostro punteggio supera i 60 punti, vi aggiudicate delle bombe "F" extra.

Se il vostro punteggio finale è un nuovo record da quando il programma è stato inizialmente caricato, potete inserire le vostre iniziali tramite il joystick. Muovendolo a destra e a sinistra, potrete spostarvi attraverso le lettere dell'alfabeto, in avanti ed indietro. Quando trovate la lettera giusta, potete selezionarla con il pulsante di sparo. Fate attenzione a non tenere premuto troppo a lungo il pulsante di sparo quando selezionate le vostre iniziali, altrimenti potreste scegliere inavvertitamente le lettere sbagliate.

La versione di "Cylon Zap" per il VIC 20 è in tre parti. La prima parte ridefinisce la locazione di partenza dei programmi BASIC per creare spazio per un set di caratteri ridefinito. Carica anche la seconda parte, che inserisce in memoria i caratteri speciali e permette di leggere le istruzioni. La terza parte contiene il gioco vero e proprio. Ogni parte deve essere battuta separatamente e quindi salvata su nastro. Se vengono caricate in sequenza, il gioco verrà caricato senza intoppi. Se utilizzate una unità dischi, sarete costretti a caricare la seconda e la terza parte manualmente quando la scritta "Press Play On Tape" apparirà sullo schermo. La versione di "Cylon Zap" per il Commodore 64 consiste in un'unica sezione a causa della maggiore memoria disponibile per il C64. Per ogni altro aspetto il gioco è fondamentalmente identico.

Un'analisi dettagliata della versione VIC

Ecco alcuni brevi commenti alla versione VIC di Cylon Zap. Queste informazioni sono destinate ai lettori interessati alle tecniche utilizzate.

Programma 1: Preparazione (VIC 20)

- Inizializza il puntatore all'inizio del programma BASIC.
- Inizializza entrambi i puntatori alla fine della memoria.
- Inizializza l'inizio del programma BASIC a O.
- Mostra il messaggio sui byte disponibili.
- Inserisce una istruzione LOAD nel buffer della tastiera.

Programma 2: Istruzioni speciali (VIC 20)

- | | |
|---------|--------------------------------------------------------------------------------------------------------------------------------------|
| 30 | Pulisce lo schermo, inizializza i colori video, entra in minuscolo |
| 35 | Uscita del titolo |
| 40-70 | Carica i caratteri |
| 75 | Carica la routine in linguaggio macchina, pone la domanda circa le istruzioni |
| 90 | Inserisce tramite una istruzione POKE un comando LOAD nel buffer della tastiera, pulisce lo schermo. Istruzioni DATA per i caratteri |
| 95-125 | Istruzioni |
| 130-160 | Istruzione di RETURN per fra proseguire il messaggio |
| 165 | Istruzioni, RETURN per dare il comando di LOAD |
| 170-180 | Sottoprogramma per il messaggio "Premi RETURN" |
| 190-230 | Titolo di apertura |
| 235-290 | Routine per muovere il titolo attraverso lo schermi |
| 300-310 | Routine di inserimento del linguaggio macchina. |
| 400-513 | |

Programma 3: Cylon Zap (VIC 20)

- | | |
|---------|------------------------------------------------------------------------------------------------------------|
| 35-50 | Inizializzazione delle variabili |
| 55 | Inizializzazione del colore dello schermo, pulizia dello stesso, quindi viene mostrato il punteggio record |
| 60 | Inizializzazione del puntatore al set di caratteri |
| 70-80 | Viene pulito lo schermo e viene disegnata la base |
| 90 | Mostra il punteggio, le basi e le bombe rimaste nell'angolo superiore sinistro |
| 150-225 | Il caccia nemico compare, si muove e colpisce |
| 230 | Se non rimangono altre basi, viene mostrato il punteggio |
| 233 | Punteggio di 50 o più, le astronavi si muovono più rapidamente |
| 235 | GOTO all'inizio della routine del joystick |
| 240-315 | Disegna la base |
| 320-340 | Aziona il laser superiore |
| 345-365 | Aziona il laser inferiore |
| 370-390 | Aziona il laser di sinistra |
| 395-415 | Aziona il laser di destra |
| 420-465 | Musica per l'introduzione delle iniziali |
| 470-475 | Istruzioni DATA per il motivo musicale |
| 480-525 | Esplosione per colpo a segno su di una caccia nemico, viene aumentato il punteggio |
| 530-580 | Sottoprogramma per basi e bombe extra |
| 585-615 | La base esplose |
| 620-650 | Se il punteggio è fra quelli record, riordina gli altri punteggi |
| 655-685 | Mostra i punteggi record ed offre la possibilità di giocare ancora |
| 690-725 | Grafica dell'esplosione della base |
| 730-815 | Sottoprogramma per l'introduzione delle iniziali |
| 820-850 | Mostra i punteggi ed il titolo |
| 855-895 | Esplosione delle bombe "F" e aggiornamento del punteggio |
| 900 | Allarme dell'astronave |
| 905-925 | Sottoprogramma di stampa delle iniziali inserite |

Cylon Zap versione per VIC 20 Programma 1

```
10 POKE52,56:POKE56,56:POKE44,28:POKE7168
,0:POKE631,131:POKE198,1 :rem 39
20 PRINT"{CLR}**** CBM BASIC V2 *****:PRI
NT"{SIN}"FRE(0);"BYTES FREE" :rem 0
```

Cylon Zap versione per VIC 20 Programma 2

```
30 PRINT"{CLR}":POKE36879,8:POKE36869,194
:rem 232
35 GOSUB235 :rem 130
40 PRINT"{ 3 GIU' }{RVS}CARICAMENTO
{ 2 SPAZI}CARA{ 2 T}ERI" :rem 63
```

```

45 FORA=5120TO7167:POKEA,PEEK(A+27648):NE
   XT                                     :rem 74
50 FORA=5384TO5504                       :rem 15
55 READD                                  :rem 202
60 IFD<>-1THENPOKEA,D:NEXT              :rem 153
65 FORA=5584TO5632:READD:IFD<>-1THENPOKEA
   ,D:NEXT                                :rem 32
70 FORA=5336TO5359:READD:POKEA,D:NEXT
   :rem 67
75 GOSUB400:PRINT"{SU} ISTRUZIONI{OFF}
   {RVS}S{OFF} O {RVS}N{OFF} " :rem 6
80 GETA$:IFA$=""THENPOKE38340,INT(RND(1)*
   7+1):POKE38344,INT(RND(1)*7+1):GOTO80
   :rem 28
85 IFA$="S"THENPOKE36869,192:GOSUB130
   :rem 185
90 POKE36869,192:POKE198,1:POKE631,131:PR
   INT"{ 2 GIU'}{CLR}ATTENDERE CARICAMENT
   O":END                                  :rem 47
95 DATA24,24,60,126,24,24,126,255,1,19,51
   ,255,255,51,19,1,128,200,204,255,255,2
   04,200 :rem 121
100 DATA128,255,126,24,24,126,60,24,24,24
   ,24,60,24,60,126,219,195,3,7,44,254,2
   54,44,7,3 :rem 222
105 DATA192,224,52,127,127,52,224,192,195
   ,219,126,60,24,60,24,24,16,8,16,8,16,
   8,16,8 :rem 92
110 DATA145,74,44,113,142,52,82,137,0,0,0
   ,170,85,0,0,0,-1 :rem 244
115 DATA0,0,0,117,85,117,69,71,0,0,0,87,1
   14,114,114,82,0,0,0,64,64,64,64,64
   :rem 120
120 DATA0,0,0,206,170,206,170,202,0,0,0,2
   32,136,232,40,232,0,0,0,0,0,0,0,-1
   :rem 130
125 DATA0,0,0,206,170,202,170,206,0,0,0,1
   39,218,171,138,139,0,0,0,56,160,48,16
   0,56 :rem 196
130 PRINT"{CLR}{RED}BENVENUTI A CYLON ZAP
   " :rem 117
135 PRINT"AVETE UNA BASE CHIAMA-TA ALPHA"
   :rem 169
140 PRINT"{GIU'}{CYN}LA VOSTRA MISSIONE E
   ' DI PROTEGGERE UN REAT-TORE NUCLEARE
   DALLE" :rem 189
145 PRINT"{PUR}ASTRONAVI KAMIKAZE"
   :rem 11
150 PRINT"{GIU'}{GRN}AVETE A DISPOSIZIONE
   4LASER CONTROLLATI DAL JOYSTICK"
   :rem 35
155 PRINT"{GIU'}{BLU}AVETE ANCHE BOMBE 'F
   ' INNESCADE DAL PULSANTEDI SPARO"
   :rem 82
160 GOSUB190 :rem 177
165 PRINT"{CLR}{YEL}DOVETE SOLO PUNTARE I
   LCANNONE E IL LASER FA FUOCO AUTOMATI
   CAMENTE" :rem 13
170 PRINT"{GIU'}{PUR}LE ASTRONAVI VOLERAN
   NOTANTO PIU' RAPIDAMENTE" :rem 215
173 PRINT"{SU}QUANTE PIU' NE VERRAN-NO DI
   STRUTTE" :rem 211
175 PRINT"{GIU'}{YEL}BASI E BOMBE EXTRA P
   ERCHI RAGGIUNGE 60 PUNTI" :rem 180
180 PRINT"{BLU}{GIU'}{ 4 SPAZI}{RVS}BUONA
   FORTUNA":GOSUB190:RETURN :rem 174
190 A$="{RVS}" :rem 146
195 FORL=1TO1000 :rem 164
200 PRINT"{HOME}" :rem 118
205 PRINTTAB(2)A$;"{CYN}{ 17 GIU'}
   { 2 SIN}RETURN PER CONTINUARE"
   :rem 27

```

```

210 GETR$:IFR$=CHR$(13) THENRETURN
   :rem 42
215 FORI=1TO333:NEXT :rem 235
220 IFA$="{RVS}"THENAS="{OFF}":GOTO230
   :rem 202
225 IFA$="{OFF}"THENAS="{RVS}":GOTO230
   :rem 207
230 NEXTL :rem 32
235 A$="{RED}*** * * *{ 3 SPAZI}*** *
   { 2 SPAZI}*":X=LEN(A$):Z$="{GIU'}":GO
   SUB295 :rem 114
240 A$="*{ 3 SPAZI}* * *{ 3 SPAZI}* * *
   *":X=LEN(A$):Z$="{ 2 GIU'}":GOSUB295
   :rem 15
245 A$="*{ 4 SPAZI}*{ 2 SPAZI}*
   { 3 SPAZI}* * * *":X=LEN(A$):Z$="
   { 3 GIU'}":GOSUB295 :rem 251
250 A$="***{ 2 SPAZI}*{ 2 SPAZI}*** * * *
   { 2 SPAZI}*":X=LEN(A$):Z$="{ 4 GIU'}
   ":GOSUB295 :rem 176
255 A$="{YEL}{ 2 SPAZI}*** * * *
   { 2 SPAZI}* * { 3 SPAZI}":X=LEN(A$):Z$
   ="{ 7 GIU'}":GOSUB295 :rem 92
260 A$="{ 4 SPAZI}* * * * { 2 SPAZI}* *
   { 3 SPAZI}":X=LEN(A$):Z$="{ 8 GIU'}":
   GOSUB295 :rem 35
265 A$="{ 3 SPAZI}*{ 2 SPAZI}*** * *
   { 2 SPAZI}* * { 3 SPAZI}":X=LEN(A$):Z$
   ="{ 9 GIU'}":GOSUB295 :rem 141
270 A$="{ 2 SPAZI}*{ 3 SPAZI}* * *
   { 10 SPAZI}":X=LEN(A$):Z$="
   { 10 GIU'}":GOSUB295 :rem 200
275 A$="{ 2 SPAZI}*** * * *{ 4 SPAZI}* *
   { 3 SPAZI}":X=LEN(A$):Z$="{ 11 GIU'}
   ":GOSUB295 :rem 134
280 PRINT:PRINT :rem 238
290 GOTO40 :rem 56
295 POKE36878,15 :rem 113
300 FORI=1TOLEN(A$) :rem 105
305 PRINT"{HOME}{GIU'}"Z$;SPC(X)LEFT$(A$,
   I):POKE36876,255-(I*7) :rem 182
310 X=X-1:NEXT:POKE36876,0:RETURN
   :rem 69
400 FORA=4700TO5000 :rem 44
420 READD :rem 246
430 POKEA,D :rem 119
440 NEXT :rem 215
450 RETURN :rem 121
500 DATA169,8,141,15,144,169,147,32,210,2
   55,162,8,160,8,32,240,255,169,18,32,2
   10,255,169 :rem 40
501 DATA169,32,210,255,169,127,32,210,255
   ,169,146,32,210,255,169,32,32,210,255
   ,169,18,32 :rem 36
502 DATA210,255,169,169,32,210,255,169,12
   7,32,210,255,24,162,9,160,7,32,240,25
   5,169,169 :rem 251
503 DATA32,210,255,169,160,162,5,32,210,2
   55,202,224,0,208,248,169,127,32,210,2
   55,24 :rem 23
504 DATA162,10,160,7,32,240,255,169,146,3
   2,210,255,169,127,32,210,255,169,18,3
   2,210,255 :rem 232
505 DATA169,160,162,5,32,210,255,202,224,
   0,208,248,169,146,32,210,255,169,169,
   32,210,255 :rem 32
506 DATA24,162,11,160,7,32,240,255,169,32
   ,32,210,255,169,18,32,210,255,169,160
   ,162,5,32 :rem 227
507 DATA210,255,202,224,0,208,248,169,146
   ,32,210,255,169,32,32,210,255,24,162,
   11,160,7 :rem 172

```

```

508 DATA32,240,255,169,18,32,210,255,24,1
    62,12,160,7,32,240,255,169,169,32,210
    ,255,169 :rem 194
509 DATA160,162,5,32,210,255,202,224,0,20
    8,248,169,127,32,210,255,24,162,13,16
    0,7,32,240 :rem 7
510 DATA255,169,146,32,210,255,169,127,32
    ,210,255,169,18,32,210,255,169,160,16
    2,5,32,210 :rem 31
511 DATA255,202,224,0,208,248,169,146,32,
    210,255,169,169,32,210,255,24,169,146
    ,32,210 :rem 141
512 DATA255,24,162,14,160,8,32,240,255,16
    9,127,32,210,255,169,169,32,210,255,1
    69,32,32 :rem 192
513 DATA210,255,169,127,32,210,255,169,16
    9,32,210,255,24,96 :rem 149

```

Cylon Zap versione per VIC 20 Programma 3

```

35 DEFFNA (A) =INT (RND (1) *X+A) :TT=4348
    :rem 93
40 V3=36876:N1=4106:N2=4590:N3=4338:N4=43
    59:V1=36876 :rem 102
45 CS=36879:S0=36878:C=33792:V4=36877:W1=
    30:W2=20:W3=10:W4=5:W5=1 :rem 186
50 A1$="D.." :A2$="U.." :A3$="D.." :A4$="C.."
    :A5$="O.." :rem 5
55 POKECS,8:PRINT "{CLR}":GOTO655:rem 68
60 BASE=3:S1=1:S2=1:S3=1:S4=1:BOM=3:SC=0
    :rem 108
65 POKE36869,205:X=15:Y=1:I=22 :rem 199
70 PRINT "{CLR} {WHT}":POKECS,8 :rem 51
75 DD=37154:P1=37151:P2=37152 :rem 41
80 GOSUB240 :rem 126
90 PRINT "{HOME} {CYN}";;<"SC:PRINT "{HOME}
    {GIU' }=>?"BA:PRINT "[£]"BOM :rem 156
95 POKEDD,127:P=PEEK (P2) AND128 :rem 24
100 J0=- (P=0) :rem 131
105 POKEDD,255:P=PEEK (P1) :rem 211
110 J1=- ( (PAND8) =0) :J2=- ( (PAND16) =0) :J3=-
    ( (PAND4) =0) :FB=- ( (PAND32) =0) :G=42
    :rem 8
115 POKETT,102 :rem 45
120 POKETT+C,INT (RND (1) *7+1) :rem 99
125 IFJ3THEN320 :rem 104
130 IFJ1THEN345 :rem 105
135 IFJ2THEN370 :rem 109
140 IFJ0THEN395 :rem 110
145 IFFBANDBOM>0THEN855 :rem 161
150 A1=FNA (1) :rem 156
155 A2=FNA (2) :rem 163
160 A3=FNA (3) :rem 161
165 A4=FNA (4) :rem 168
170 IFA1=1ANDS1<>0THENS1=0:GOSUB900
    :rem 123
175 IFA2=2ANDS2<>0THENS2=0:GOSUB900
    :rem 132
180 IFA3=3ANDS3<>0THENS3=0:GOSUB900
    :rem 132
185 IFA4=4ANDS4<>0THENS4=0:GOSUB900
    :rem 141
190 IFS1=0ANDPEEK (N1+22) <>102THENN1=N1+I:
    POKEN1+C,4:POKEN1,40:POKEN1-I,32
    :rem 121
195 IFPEEK (N1+22) =102THENGOSUB585
    :rem 211
200 IFS2=0ANDPEEK (N2-22) <>102THENN2=N2-I:
    POKEN2+C,3:POKEN2,37:POKEN2+I,32
    :rem 127

```

LOAD

K

Cylon Zap

```

205 IFPEEK (N2-22) =102THENGOSUB585
    :rem 206
210 IFS3=0ANDPEEK (N3+1) <>102THENN3=N3+Y:P
    OKEN3+C,5:POKEN3,39:POKEN3-Y,32
    :rem 118
215 IFPEEK (N3+1) =102THENGOSUB585:rem 155
220 IFS4=0ANDPEEK (N4-1) <>102THENN4=N4-Y:P
    OKEN4+C,6:POKEN4,38:POKEN4+Y,32
    :rem 128
225 IFPEEK (N4-1) =102THENGOSUB585:rem 159
230 IFBASE=0THENGOTO620 :rem 172
233 IFSC>50THENX=4 :rem 88
235 GOTO90 :rem 60
240 PRINT "{RED}":SYS4700 :rem 135
295 POKE4282+C,7:POKE4282,33:POKE4414+C,7
    :POKE4414,36:POKE4345+C,7:POKE4345,34
    :rem 227
300 POKE4351+C,7:POKE4351,35 :rem 42
310 POKETT-1,102:POKETT+1,102:POKETT-22,1
    02:POKETT+22,102 :rem 116
315 RETURN :rem 121
320 POKEV4,245:POKES0,15 :rem 147
325 FORF=4282TO4106STEP-22 :rem 15
330 IFPEEK (F-22) <>40THENPOKEF+C,7:POKEF,4
    1:FORT=1TO5:NEXT:POKEF,32:NEXT
    :rem 242
335 IFPEEK (F-22) =40THENPOKEN1+C,2:POKEN1,
    42:GOSUB480:POKEN1,32:N1=4106:S1=1
    :rem 247
340 POKEV4,0:POKE4282,33:GOTO150:rem 128
345 POKEV4,245:POKES0,15 :rem 154
350 FORF=4414TO4602STEP22 :rem 222
355 IFPEEK (F+22) <>37THENPOKEF+C,7:POKEF,4
    1:FORT=1TO5:NEXT:POKEF,32:NEXT
    :rem 253
360 IFPEEK (F+22) =37THEN:POKEN2+C,2:POKEN2
    ,42:GOSUB480:POKEN2,32:N2=4590:S2=1
    :rem 63
365 POKEV4,0:POKE4414,36:GOTO150:rem 135
370 POKES0,15:POKEV4,245 :rem 152
375 FORF=4345TO4338STEP-1 :rem 232
380 IFPEEK (F-1) <>39THENPOKEF+C,7:POKEF,43
    :FORT=1TO5:NEXT:POKEF,32:NEXT
    :rem 206
385 IFPEEK (F-1) =39THENPOKEN3+C,2:POKEN3,4
    2:GOSUB480:POKEN3,32:N3=4338:S3=1
    :rem 226
390 POKEV4,0:POKE4345,34:GOTO150:rem 134
395 POKES0,15:POKEV4,245 :rem 159
400 FORF=4351TO4359 :rem 67
405 IFPEEK (F+1) <>38THENPOKEF+C,7:POKEF,43
    :FORT=1TO5:NEXT:POKEF,32:NEXT
    :rem 201
410 IFPEEK (F+1) =38THENPOKEN4+C,2:POKEN4,4
    2:GOSUB480:POKEN4,32:N4=4359:S4=1
    :rem 220
415 POKEV4,0:POKE4351,35:GOTO150:rem 130
420 POKES0,15:RESTORE :rem 56
425 READP :rem 7
430 IFP=-1THEN465 :rem 223
435 READD :rem 252
440 POKEV3,P:POKEV1,P :rem 56

```

```

445 FORN=1TOD:NEXT :rem 160
450 POKEV3,0:POKEV1,0 :rem 249
455 FORN=1TO20:NEXT :rem 191
460 GOTO425 :rem 110
465 RETURN :rem 127
470 DATA217,200,213,200,223,200,227,100,2
34,100,230,200 :rem 147
475 DATA227,100,234,100,230,200,223,200,2
27,200,217,200,213,300,-1 :rem 168
480 POKES0,15 :rem 224
485 POKEV4,200 :rem 24
490 FORL=15TO0STEP-1 :rem 224
495 POKES0,L :rem 204
500 NEXT:POKEV4,0 :rem 35
505 SC=SC+1 :rem 95
510 IFSC=30THENX=INT(X/2):Y=2 :rem 22
515 IFSC=50THENX=4:I=44:BOM=BOM+1
:rem 17
520 IFSC=60ORSC=110ORSC=150THENGOTO530
:rem 111
525 RETURN :rem 124
530 PRINT"{CLR}{ 10 GIU'}{ 9 SPAZI}BONUS"
:rem 45
535 PRINT"{ 6 SPAZI}BASI - BOMBE"
:rem 31
540 POKEV4,0:POKE36878,15:L=0 :rem 172
545 FORT=1TO10 :rem 75
550 POKE36876,220 :rem 151
555 NEXT :rem 222
560 FORT=1TO10 :rem 72
565 POKE36876,230 :rem 158
570 NEXT :rem 219
575 IFL<6THENL=L+1:GOTO545 :rem 95
580 BOM=BOM+1:BASE=BASE+1:SC=SC+5:POKE368
76,0:POKE36878,0:PRINT"{CLR}":GOSUB24
0:GOTO510 :rem 157
585 POKES0,15:Q1=4348:Q2=4349:Q3=4370:Q4=
4371:K=0:Q5=Q1-23:Q6=Q3+23:Q7=Q1+21:Q
8=4372 :rem 8
590 POKEV4,200:KK=8 :rem 90
595 FORZ=15TO0STEP-2 :rem 245
600 POKES0,Z:GOSUB690:NEXT:POKECS,8:POKEV
4,0 :rem 82
605 N1=4106:S1=1:N2=4590:S2=1:N3=4338:S3=
1:N4=4359:S4=1:PRINT"{CLR}" :rem 215
610 BASE=BASE-1:IFBASE<>0THENGOSUB240
:rem 59
615 RETURN :rem 124
620 POKE36869,192:PRINT"{CLR}" :rem 61
625 IFSC=>W1THENA5$=A4$:A4$=A3$:A3$=A2$
:rem 178
630 IFSC=>W1THENA2$=A1$:W5=W4:W4=W3:W3=W2
:W2=W1:W1=SC:GOTO905 :rem 44
635 IFSC=>W2ANDSC<W1THENA5$=A4$:A4$=A3$:A
3$=A2$:W5=W4:W4=W3:W3=W2:W2=SC:GOTO91
0 :rem 40
640 IFSC=>W3ANDSC<W2THENA5$=A4$:A4$=A3$:W
5=W4:W4=W3:W3=SC:GOTO915 :rem 252
645 IFSC=>W4ANDSC<W3THENA5$=A4$:W5=W4:W4=
SC:GOTO920 :rem 204
650 IFSC=>W5ANDSC<W4THENW5=SC:GOTO925
:rem 152
655 GOSUB820:PRINT"{HOME}{YEL}
{ 21 GIU'}PER GIOCARE PREMI {RVS}
{CYN}S" :rem 61
660 GETZ$:IFZ$=""THENFORCC=37904TO37907:P
OKECC,INT(RND(1)*7+1):NEXT :rem 66
665 POKE38369,INT(RND(1)*7+1) :rem 104
670 IFZ$=""THEN660 :rem 246
675 IFZ$="S"THEN60 :rem 24
680 IFZ$="N"THENPRINT"{CLR}{BLU}":POKECS,
27:END :rem 209
685 GOTO655 :rem 124
690 K=K+1:M=23:N=22:O=21:R=INT(RND(1)*7+1
):IFK>3ANDKK<110THENPOKECS,KK:KK=KK+1
7 :rem 25
695 POKEQ1,G:POKEQ2,G:POKEQ3,G:POKEQ4,G:P
OKEQ5,G:POKEQ6,G:POKEQ7,G:POKEQ8,G
:rem 118
700 POKEQ1+C,R:POKEQ2+C,INT(RND(1)*7+1):P
OKEQ3+C,R:POKEQ4+C,INT(RND(1)*7+1)
:rem 217
705 POKEQ5+C,R:POKEQ6+C,INT(RND(1)*7+1):P
OKEQ7+C,R:POKEQ8+C,INT(RND(1)*7+1)
:rem 238
710 FORT=1TO10:NEXT :rem 190
715 IFK>3THENG=46:PRINT"{CLR}" :rem 163
720 IFK<8THENQ1=Q1-O:Q2=Q2-M:Q3=Q3+O:Q4=Q
4+M:Q5=Q5-N:Q6=Q6+N:Q7=Q7-1:Q8=Q8+1:R
ETURN :rem 156
725 PRINT"{CLR}":RETURN :rem 28
730 PRINT"{ 3 GIU'}":CH=4171:E=1:rem 90
735 POKEDD,127:P=PEEK(P2)AND128 :rem 73
740 J0=- (P=0) :rem 141
745 POKEDD,255:P=PEEK(P1) :rem 221
750 J2=- ((PAND16)=0) :rem 27
755 FB=- ((PAND32)=0) :rem 42
760 IFJ0THENE=E+1 :rem 248
765 IFJ2THENE=E-1 :rem 1
770 IFE=0THENE=26 :rem 248
775 IFE=27THENE=1 :rem 255
780 POKECH,E:POKECH+C,7 :rem 143
785 FORT=1TO100:NEXT :rem 250
790 POKECH+C,6 :rem 42
795 IFFBANDCH=4171THENN1$=CHR$(E+64):CH=C
H+1:E=1:GOTO735 :rem 71
800 IFFBANDCH=4172THENN2$=CHR$(E+64):CH=C
H+1:E=1:GOTO735 :rem 60
805 IFFBANDCH=4173THENN3$=CHR$(E+64):CH=C
H+1:E=32:GOTO735 :rem 119
810 IFCH=4174THENN5$=N1$+N2$+N3$:RETURN
:rem 47
815 GOTO735 :rem 118
820 PRINT"{CLR}{BLU}{ 2 SPAZI}EROI DI CYL
ON ZAP":PRINT:PRINT"{RED}I MIGLIORI 5
PUNTEGGI{OFF}" :rem 6
825 PRINT"{HOME}{GIU'}{CYN}{ 4 GIU'}
{ 6 SPAZI}"A1$"...W1 :rem 99
830 PRINT"{PUR}{ 2 GIU'}{ 6 SPAZI}"A2$"..
.W2 :rem 24
835 PRINT"{PUR}{ 2 GIU'}{ 6 SPAZI}"A3$"..
.W3 :rem 31
840 PRINT"{PUR}{ 2 GIU'}{ 6 SPAZI}"A4$"..
.W4 :rem 29
845 PRINT"{PUR}{ 2 GIU'}{ 6 SPAZI}"A5$"..
.W5 :rem 36
850 RETURN :rem 125
855 POKES0,15:POKEV4,220 :rem 153
860 FORCO=127TO8STEP-17 :rem 154
865 POKECS,CO:POKES0,CO/8-1 :rem 149
870 FORT=1TO100:NEXT:NEXTCO :rem 0
875 IFS1=0THENSC=SC+1:GOSUB510:POKEN1,32:
N1=4106:S1=1 :rem 206
880 IFS2=0THENSC=SC+1:GOSUB510:POKEN2,32:
N2=4590:S2=1 :rem 213
885 IFS3=0THENSC=SC+1:GOSUB510:POKEN3,32:
N3=4338:S3=1 :rem 222
890 IFS4=0THENSC=SC+1:GOSUB510:POKEN4,32:
N4=4359:S4=1 :rem 225
895 POKEV4,0:BOM=BOM-1:GOTO150 :rem 85
900 POKES0,15:FORM=180TO235STEP2:POKEV3,M
:NEXT:POKEV3,0:POKES0,0:RETURN
:rem 96
905 PRINT"{HOME} #1 INTRODUCI LE TUE

```

```

{ 3 SPAZI}INIZIALI":GOSUB420:GOSUB730
:A1$=N5$:GOTO655 :rem 202
910 PRINT"{HOME} #2 INTRODUCI LE TUE
{ 3 SPAZI}INIZIALI":GOSUB420:GOSUB730
:A2$=N5$:GOTO655 :rem 200
915 PRINT"{HOME} #3 INTRODUCI LE TUE
{ 3 SPAZI}INIZIALI":GOSUB420:GOSUB730
:A3$=N5$:GOTO655 :rem 207
920 PRINT"{HOME} #4 INTRODUCI LE TUE
{ 3 SPAZI}INIZIALI":GOSUB420:GOSUB730
:A4$=N5$:GOTO655 :rem 205
925 PRINT"{HOME} #5 INTRODUCI LE TUE
{ 3 SPAZI}INIZIALI":GOSUB420:GOSUB730
:A5$=N5$:GOTO655 :rem 212

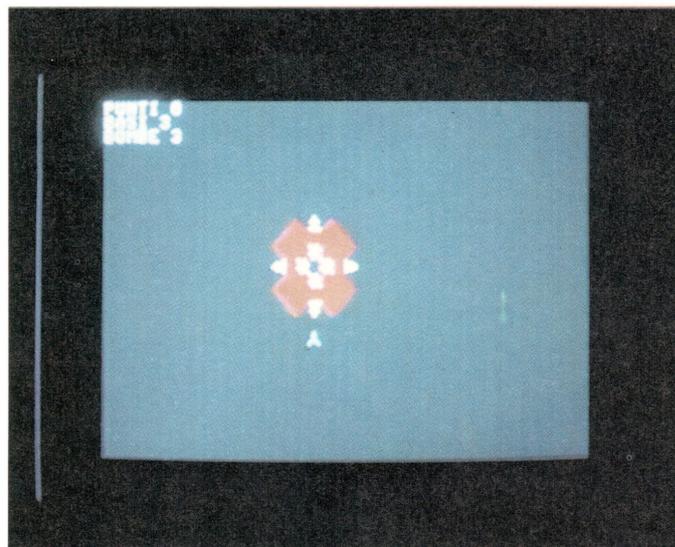
```

Cylon Zap versione per C64

```

100 POKE52,48:POKE56,48:CLR :rem 70
125 DATA28,149,100,25,30,100,33,135,100,3
7,162,50,50,60,50 :rem 54
130 DATA42,62,100,37,162,50,50,60,50,42,6
2,100,33,135,100 :rem 252
140 DATA28,49,100,25,30,100 :rem 78
145 FORX=1TO36:READRT:NEXT :rem 200
150 PRINT"{CLR}":POKE53281,0:POKE53280,0:
PRINTCHR$(14) :rem 7
160 GOSUB590 :rem 181
170 PRINT"{ 3 GIU'}{ 10 SPAZI}{RVS}CARIC
AMENTO {SPAZI}CARA{ 2 T}ERI":rem 19
180 POKE56334,(PEEK(56334)AND254):POKE1,P
EEK(1)AND251 :rem 11
190 FORA=0TO2047:POKE(A+12288),PEEK(A+532
48):NEXT :rem 87
200 FORA=12552TO12672 :rem 155
210 READD :rem 243
220 IFD<>-1THENPOKEA,D:NEXT :rem 199
230 FORA=12288TO14335:READD:IFD<>-1THENPO
KEA,PEEK(A):NEXT :rem 28
240 FORA=12504TO12527:READD:POKEA,D:NEXT
:rem 200
250 POKE1,55 :rem 141
260 POKE56334,PEEK(56334)OR1 :rem 69
270 GOSUB750:PRINT"{SU}{ 9 SPAZI}ISTRUZI
ONI{OFF} {RVS}S{OFF} O {RVS}N{OFF}
{ 11 SPAZI}" :rem 59
280 GETA$:IFA$=""THENPOKE56076,INT(RND(1)
*7+1):POKE56080,INT(RND(1)*7+1):GOTO2
80 :rem 131
290 IFA$="S"THENPOKE53272,(PEEK(53272)AND
240)+12:GOSUB380 :rem 6
300 GOTO1000 :rem 141
310 DATA24,24,60,126,24,24,126,255,1,19,5
1,255,255,51,19,1,128,200,204,255,255
:rem 31
315 DATA204,200 :rem 7
320 DATA128,255,126,24,24,126,60,24,24,24
,24,60,24,60,126,219,195,3,7,44,254
:rem 197
325 DATA254,44,7,3,192,224,52,127,127,52,
224,192,195,219,126,60,24,60,24,24
:rem 161
330 DATA16,8,16,8,16,8,16,8,145,74,44,113
,142,52,82,137,0,0,0,170,85,0,0,0,-1
:rem 212
350 DATA0,0,0,119,68,116,20,119,0,0,0,119
,85,87,86,117,0,0,0,112,64,96,64,112
:rem 221
360 DATA0,0,0,206,170,206,170,202,0,0,0,2
38,136,236,40,238,0,0,0,224,128,224
:rem 145

```



```

365 DATA32,224,-1,0,0,0 :rem 127
370 DATA206,170,202,170,206,0,0,0,139,218
,171,138,139,0,0,0,56,160,56,136,56
:rem 180
380 PRINT"{CLR}{RED}BENVENUTI A CYLON ZAP
" :rem 124
390 PRINT"AVETE UNA BASE CHIAMATA ALPHA":
PRINT :rem 70
400 PRINT"{CYN}LA VOSTRA MISSIONE CONSIST
E NEL PROTEG- GERE UN{ 2 SPAZI}REATTO
RE NUCLEARE"; :rem 210
410 PRINT" DALLE {PUR}ASTRO- NAVI KAMIKAZ
E" :rem 149
420 PRINT"{GIU'}{GRN}AVETE A DISPOSIZIONE
4 LASER CONTROLLATIDAL JOYSTICK"
:rem 35
430 PRINT"{BLU}AVETE ANCHE BOMBE F INNESC
ATE DAL PUL-{ 2 SPAZI}SANTE DI SPARO"
:rem 28
440 PRINT"{GIU'}{YEL}DOVETE SOLO PUNTARE
IL CANNONE E IL LA- SER SPARA AUTOMAT
ICAMENTE" :rem 40
450 GOSUB500 :rem 174
460 PRINT"{CLR}{PUR}{GIU'}LE ASTRONAVI VO
LERANNO TANTO PIU RAPIDA-" :rem 249
465 PRINT"{SU}MENTE QUANTE PIU NE DISTRUG
GERETE" :rem 167
470 PRINT"{GIU'}{YEL}BASI E BOMBE EXTRA P
ER CHI RAGGIUNGE 60 PUNTI" :rem 178
480 PRINT"{BLU}{GIU'}{ 7 SPAZI}{RVS}BUONA
FORTUNA{OFF}":GOSUB500:RETURN
:rem 62
490 GOTO65535 :rem 222
500 A$="{RVS}" :rem 141
510 FORL=1TO1000 :rem 155
520 PRINT"{HOME}" :rem 123
530 PRINTTAB(2)A$;"{CYN}{ 20 GIU'}PREMI R
ETURN PER CONTINUARE" :rem 146
540 GETR$:IFR$=CHR$(13) THENRETURN
:rem 48
550 FORI=1TO333:NEXT :rem 237
560 IFA$="{RVS}"THENA$="{OFF}":GOTO580
:rem 217
570 IFA$="{OFF}"THENA$="{RVS}":GOTO580
:rem 218
580 NEXTL :rem 40
590 A$="{RED}*** *{ 3 SPAZI}* *
{ 4 SPAZI}***{ 2 SPAZI}*{ 2 SPAZI}*":
X=LEN(A$):Z$="{GIU'}":GOSUB710
:rem 110

```

```

600 A$="*{ 4 SPAZI}* *{ 2 SPAZI}*
    { 4 SPAZI}* *{ 2 SPAZI}** *":X=LEN(A$
    ):Z$="{ 2 GIU}":GOSUB710 :rem 7
610 A$="*{ 5 SPAZI}*{ 3 SPAZI}*
    { 4 SPAZI}* *{ 2 SPAZI}* *":X=LEN(A$
    ):Z$="{ 3 GIU}":GOSUB710 :rem 239
611 A$="*{ 5 SPAZI}*{ 3 SPAZI}*
    { 4 SPAZI}* *{ 2 SPAZI}*{ 2 SPAZI}**":
    X=LEN(A$):Z$="{ 4 GIU}":GOSUB710
    :rem 215
620 A$="***{ 3 SPAZI}*{ 3 SPAZI}***
    { 2 SPAZI}***{ 2 SPAZI}*{ 2 SPAZI}**":
    X=LEN(A$):Z$="{ 5 GIU}":GOSUB710
    :rem 186
630 A$="{YEL}{ 3 SPAZI}***{ 2 SPAZI}***
    { 2 SPAZI}***{ 2 SPAZI}* *{ 2 SPAZI}
    "":X=LEN(A$):Z$="{ 8 GIU}":GOSUB710
    :rem 98
640 A$="{ 5 SPAZI}*{ 2 SPAZI}* *
    { 2 SPAZI}* *{ 2 SPAZI}* *{ 3 SPAZI}
    "":X=LEN(A$):Z$="{ 9 GIU}":GOSUB710
    :rem 46
650 A$="{ 4 SPAZI}*{ 3 SPAZI}***
    { 2 SPAZI}***{ 2 SPAZI}* *{ 3 SPAZI}
    "":X=LEN(A$):Z$="{ 10 GIU}":GOSUB710
    :rem 148
660 A$="{ 3 SPAZI}*{ 4 SPAZI}* *
    { 2 SPAZI}*{ 10 SPAZI}":X=LEN(A$):Z$=
    "{ 11 GIU}":GOSUB710 :rem 212
670 A$="{ 3 SPAZI}***{ 2 SPAZI}* *
    { 2 SPAZI}*{ 4 SPAZI}* *{ 3 SPAZI}":X
    =LEN(A$):Z$="{ 12 GIU}":GOSUB710
    :rem 142
680 PRINT:PRINT :rem 242
700 GOTO170 :rem 104
710 S=54272 :rem 44
711 POKE54296,15:POKE54277,18:POKE54278,2
    40 :rem 160
712 POKE54276,33 :rem 99
720 FORI=1TOLEN(A$):POKE54273,I+40
    :rem 225
721 PRINT"{HOME}{GIU}">{ 8 DES}"Z$;SPC(X)L
    EFT$(A$,I):POKE54272,(I*2)+180
    :rem 140
730 X=X-1:NEXT:FORG=15TO0STEP-1:POKE54296
    ,G:NEXT:POKES+4,16 :rem 227
735 FORE=STOS+28:POKEE,0:NEXT:RETURN
    :rem 131
750 FORA=49152TO49453 :rem 178
760 READD :rem 253
770 POKEA,D :rem 126
780 NEXT :rem 222
790 RETURN :rem 128
800 DATA169,12,141,33,208,169,147,32,210,
    255,162,8,160,16,32,240,255,169,18,32
    :rem 51
810 DATA210,255,169 :rem 218
820 DATA169,32,210,255,169,127,32,210,255
    ,169,146,32,210,255,169,32,32,210
    :rem 110
825 DATA255,169,18,32 :rem 71
830 DATA210,255,169,169,32,210,255,169,12
    7,32,210,255,24,162,9,160,15,32
    :rem 12
835 DATA240,255,169,169 :rem 176
840 DATA32,210,255,169,160,162,5,32,210,2
    55,202,224,0,208,248,169,127:rem 113
845 DATA32,210,255,24 :rem 57
850 DATA162,10,160,15,32,240,255,169,146,
    32,210,255,169,127,32,210,255
    :rem 162
855 DATA169,18,32,210,255 :rem 9
860 DATA169,160,162,5,32,210,255,202,224,
    0,208,248,169,146,32,210,255:rem 116
865 DATA169,169,32,210,255 :rem 65
870 DATA24,162,11,160,15,32,240,255,169,3
    2,32,210,255,169,18,32,210 :rem 8
875 DATA255,169,160,162,5,32 :rem 160
880 DATA210,255,202,224,0,208,248,169,146
    ,32,210,255,169,32,32,210,255,24
    :rem 55
885 DATA24,162,11,160,7 :rem 158
890 DATA32,240,255,169,18,32,210,255,24,1
    62,12,160,15,32,240,255,169,169
    :rem 17
895 DATA32,210,255,169 :rem 120
900 DATA160,162,5,32,210,255,202,224,0,20
    8,248,169,127,32,210,255,24 :rem 52
905 DATA162,13,160,15,32,240 :rem 137
910 DATA255,169,146,32,210,255,169,127,32
    ,210,255,169,18,32,210,255 :rem 30
915 DATA169,160,162,5,32,210 :rem 146
920 DATA255,202,224,0,208,248,169,146,32,
    210,255,169,169,32,210,255,24
    :rem 174
925 DATA169,146,32,210 :rem 113
930 DATA255,24,162,14,160,16,32,240,255,1
    69,127,32,210,255,169,169,32:rem 126
935 DATA210,255,169,32,32 :rem 4
940 DATA210,255,169,127,32,210,255,169,16
    9,32,210,255,24,96 :rem 153
1000 RESTORE:CLR :rem 0
1060 DEFFNA(A)=INT(RND(1)*X+A):TT=1482
    :rem 184
1070 POKE53272,(PEEK(53272)AND240)+12
    :rem 231
1080 N1=1042:N2=1922:N3=1464:N4=1502:V1=3
    6876 :rem 165
1090 CS=53281:C=54272:W1=30:W2=20:W3=10:W
    4=5:W5=1 :rem 239
1100 A1$="D..":A2$="U..":A3$="D..":A4$="C
    ..":A5$="O.." :rem 98
1110 POKECS,1:PRINT"{CLR}":GOTO2190
    :rem 194
1120 BASE=3:S1=1:S2=1:S3=1:S4=1:BOM=3:SC=
    0 :rem 202
1130 POKECS,12:X=15:Y=1:I=40 :rem 115
1140 PRINT"{CLR}{WHT}":POKECS,8 :rem 146
1150 GOSUB1450 :rem 17
1160 PRINT"{HOME}{WHT}PUNTI"SC:PRINT"
    {HOME}{GIU}'BASI"BA:PRINT"{WHT}BOMBE
    "BO :rem 174
1170 J0=15-(PEEK(56321)AND15) :rem 20
1180 G=42:FB=(PEEK(56321)AND16) :rem 181
1190 POKETT,102 :rem 97
1200 POKETT+C,INT(RND(1)*7+1) :rem 147
1210 IFJ0=1THEN1510 :rem 49
1220 IFJ0=2THEN1570 :rem 57
1230 IFJ0=4THEN1630 :rem 57
1240 IFJ0=8THEN1690 :rem 68
1250 IFFB=0ANDBOM>0THEN2590 :rem 106
1260 A1=FNA(1) :rem 207
1270 A2=FNA(2) :rem 210
1280 A3=FNA(3) :rem 213
1290 A4=FNA(4) :rem 216
1300 IFA1=1ANDS1<>0THENS1=0:GOSUB2680
    :rem 222
1310 IFA2=2ANDS2<>0THENS2=0:GOSUB2680
    :rem 227
1320 IFA3=3ANDS3<>0THENS3=0:GOSUB2680
    :rem 232
1330 IFA4=4ANDS4<>0THENS4=0:GOSUB2680

```



Cylon Zap

```

:rem 237
1340 IFS1=0ANDPEEK(N1+40)<>102THENN1=N1+I
:POKEN1+C,4:POKEN1,40:POKEN1-I,32
:rem 167
1350 IFPEEK(N1+40)=102THENGOSUB2050
:rem 34
1360 IFS2=0ANDPEEK(N2-40)<>102THENN2=N2-I
:POKEN2+C,3:POKEN2,37:POKEN2+I,32
:rem 183
1370 IFPEEK(N2-40)=102THENGOSUB2050
:rem 39
1380 IFS3=0ANDPEEK(N3+1)<>102THENN3=N3+Y:
POKEN3+C,5:POKEN3,39:POKEN3-Y,32
:rem 175
1390 IFPEEK(N3+1)=102THENGOSUB2050
:rem 245
1400 IFS4=0ANDPEEK(N4-1)<>102THENN4=N4-Y:
POKEN4+C,6:POKEN4,38:POKEN4+Y,32
:rem 177
1410 IFPEEK(N4-1)=102THENGOSUB2050
:rem 241
1420 IFBASE=0THENGOTO2130 :rem 12
1430 IFSC>50THENX=4 :rem 136
1440 GOTO1160 :rem 202
1450 PRINT"{RED}":SYS49152:POKECS,11
:rem 130
1460 POKE1362+C,1:POKE1362,33:POKE1602+C,
1:POKE1602,36:POKE1479+C,1:POKE1479,
34 :rem 246
1470 POKE1485+C,1:POKE1485,35 :rem 103
1480 POKETT-1,102:POKETT+1,102:POKETT-40,
102:POKETT+40,102 :rem 173
1490 POKETT-1+C,1:POKETT+1+C,1:POKETT-40+
C,1:POKETT+40+C,1 :rem 222
1500 RETURN :rem 166
1510 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
:rem 247
1520 POKE54276,129 :rem 199
1530 FORF=1362TO1042STEP-40 :rem 54
1540 IFPEEK(F-40)<>40THENPOKEF+C,1:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
:rem 32
1550 IFPEEK(F-40)=40THENPOKEN1+C,2:POKEN1
,42:GOSUB1830:POKEN1,32:N1=1042:S1=1
:rem 83
1560 POKE54296,0:POKE1362,33:GOTO1260
:rem 100
1570 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
:rem 253
1580 POKE54276,129 :rem 205
1590 FORF=1602TO1944STEP40 :rem 23
1600 IFPEEK(F+40)<>37THENPOKEF+C,1:POKEF,
41:FORT=1TO5:NEXT:POKEF,32:NEXT
:rem 33
1610 IFPEEK(F+40)=37THENPOKEN2+C,2:POKEN2
,42:GOSUB1830:POKEN2,32:N2=1922:S2=-
39 :rem 200
1620 POKE54296,0:POKE1602,36:GOTO1260
:rem 97
1630 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
:rem 250
1640 POKE54276,129 :rem 202
1650 FORF=1479TO1464STEP-1 :rem 23
1660 IFPEEK(F-1)<>39THENPOKEF+C,1:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
:rem 250
1670 IFPEEK(F-1)=39THENPOKEN3+C,2:POKEN3,
42:GOSUB1830:POKEN3,32:N3=1464:S3=1
:rem 61

```

```

1680 POKE54296,0:POKE1479,34:GOTO1260
:rem 113
1690 POKE54296,15:POKE54273,33:POKE54272,
133:POKE54277,50:POKE54278,120
:rem 0
1700 POKE54276,129 :rem 199
1710 FORF=1485TO1502 :rem 112
1720 IFPEEK(F+1)<>38THENPOKEF+C,1:POKEF,4
3:FORT=1TO5:NEXT:POKEF,32:NEXT
:rem 244
1730 IFPEEK(F+1)=38THENPOKEN4+C,2:POKEN4,
42:GOSUB1830:POKEN4,32:N4=1502:S4=1
:rem 53
1740 POKE54296,0:POKE1485,35:GOTO1260
:rem 108
1745 FORS0=54272TO54272+28:POKES0,0:NEXT
:rem 121
1750 POKE54296,15:POKE54277,53:POKE54278,
69:POKE54276,33 :rem 175
1770 RESTORE:FORGB=1TO12:READHA,LA,DU:POK
E54273,HA:POKE54272,LA :rem 183
1780 FORT=1TODU:NEXTTT :rem 130
1790 NEXTGB:FOR S0=54272TO54272+28:POKES0,
0:NEXT :rem 123
1800 RETURN :rem 169
1810 DATA217,200,213,200,223,200,227,100,
234,100,230,200 :rem 194
1820 DATA227,100,234,100,230,200,223,200,
227,200,217,200,213,300,-1 :rem 211
1830 POKE54296,15:POKE54277,53:POKE54278,
67:POKE54276,129 :rem 226
1840 POKE54272,200:POKE54273,33 :rem 190
1850 FORL=15TO0STEP-1 :rem 17
1860 POKE54296,L :rem 128
1870 NEXT:POKE54276,0 :rem 220
1880 SC=SC+1 :rem 150
1890 IFSC=30THENX=INT(X/2):Y=2 :rem 82
1900 IFSC=50THENX=4:I=80:BOM=BOM+1
:rem 64
1910 IFSC=60ORSC=110ORSC=150THENGOTO1930
:rem 216
1920 RETURN :rem 172
1930 PRINT"{CLR}{ 10 GIU' } { 10 SPAZI }BONU
S"; :rem 157
1940 PRINT" BASI - BOMBE":L=0 :rem 67
1950 POKE54296,15:POKE54277,50:POKE54278,
167:POKE54276,17 :rem 223
1960 FORT=1TO10 :rem 125
1970 POKE54272,230:POKE54273,33 :rem 197
1980 NEXT :rem 17
1990 FORT=1TO10 :rem 128
2000 POKE54272,180:POKE54273,28 :rem 190
2010 NEXT :rem 2
2020 IFL<6THENL=L+1:GOTO1950 :rem 179
2030 FORD=54272TO54272+28:POKED,0:NEXT
:rem 239
2040 BOM=BOM+1:BA=BA+1:SC=SC+5:PRINT"
{CLR}":GOSUB1450:GOTO1890 :rem 94
2050 POKE54296,14:Q1=1482:Q2=1484:Q3=1522
:Q4=1524:K=0:Q5=Q1-41:Q6=Q3+41:Q7=Q1

```

```

+39 :rem 222
2060 Q8=1526:POKE54277,44:POKE54278,56:PO
KE54276,129 :rem 165
2070 POKE54272,200:POKE54273,34:KK=8
:rem 0
2080 FORZ=15TO0STEP-2 :rem 28
2090 POKE54296,Z:GOSUB2260:NEXT:POKECS,8:
POKE54276,0 :rem 183
2100 N1=1042:S1=1:N2=1922:S2=1:N3=1464:S3
=1:N4=1502:S4=1:PRINT"{CLR}"
:rem 231
2110 BASE=BASE-1:IFBASE<>0THENGOSUB1450
:rem 156
2120 RETURN :rem 165
2130 PRINT"{CLR}" :rem 42
2140 IFSC>=W1THENA5$=A4$:A4$=A3$:A3$=A2$
:rem 220
2150 IFSC>=W1THENA2$=A1$:W5=W4:W4=W3:W3=W
2:W2=W1:W1=SC:GOTO2730 :rem 137
2160 IFSC>=W3ANDSC<W2THENA5$=A4$:A4$=A3$:
W5=W4:W4=W3:W3=SC:GOTO2750 :rem 90
2170 IFSC>=W4ANDSC<W3THENA5$=A4$:W5=W4:W4
=SC:GOTO2760 :rem 43
2180 IFSC>=W5ANDSC<W4THENW5=SC:GOTO2770
:rem 248
2190 GOSUB2510:PRINT"{HOME}{BLK}
{ 21 GIU'}{ 12 SPAZI}PER GIOCARE PRE
MI {RVS}{BLK}S" :rem 122
2200 GETZ$:IFZ$=""THENFORCC=55321TO55324:
POKECC,INT(RND(1)*7+1):NEXT:rem 92
2210 POKE55165,INT(RND(1)*7+1) :rem 133
2220 IFZ$=""THEN2200 :rem 71
2230 IFZ$="S"THEN1120 :rem 155
2240 IFZ$="N"THENPRINT"{CLR}{BLU}":POKECS
,27:END :rem 251
2250 GOTO2190 :rem 206
2260 K=K+1:M=41:N=40:O=39:R=INT(RND(1)*7+
1) :rem 165
2270 IFK>3ANDK<110THENPOKECS,KK:KK=KK+31
:rem 234
2280 POKEQ1,G:POKEQ2,G:POKEQ3,G:POKEQ4,G:
POKEQ5,G:POKEQ6,G:POKEQ7,G:POKEQ8,G
:rem 158
2290 POKEQ1+C,R:POKEQ2+C,INT(RND(1)*7+1):
POKEQ3+C,R:POKEQ4+C,INT(RND(1)*7+1)
:rem 15
2300 POKEQ5+C,R:POKEQ6+C,INT(RND(1)*7+1):
POKEQ7+C,R:POKEQ8+C,INT(RND(1)*7+1)
:rem 23
2310 FORT=1TO10:NEXT :rem 236
2320 IFK>3THENG=46:PRINT"{CLR}" :rem 205
2330 IFK<8THENQ1=Q1-O:Q2=Q2-M:Q3=Q3+O:Q4=
Q4+M:Q5=Q5-N:Q6=Q6+N:Q7=Q7-1:Q8=Q8+1
:rem 177
2340 RETURN :rem 169
2350 PRINT"{CLR}":RETURN :rem 72
2360 PRINT"{ 3 GIU'}":CH=1160:E=1
:rem 134
2370 J0=15-(PEEK(56321)AND15) :rem 23
2380 FB=PEEK(56321)AND16 :rem 67
2390 IFJ0=8THENE=E+1 :rem 158
2400 IFJ0=4THENE=E-1 :rem 148
2410 IFE=0THENE=26 :rem 33
2420 IFE=27THENE=1 :rem 36
2430 POKECH,E:POKECH+C,7 :rem 185
2440 FORT=1TO100:NEXT :rem 32
2450 POKECH+C,1 :rem 80
2460 IFFB=0ANDCH=1160THENN1$=CHR$(E+64):C
H=CH+1:E=1:GOTO2370 :rem 3
2470 IFFB=0ANDCH=1161THENN2$=CHR$(E+64):C
H=CH+1:E=1:GOTO2370 :rem 6
2480 IFFB=0ANDCH=1162THENN3$=CHR$(E+64):C

```

LOAD 

Cylon Zap

```

H=CH+1:E=32:GOTO2370 :rem 61
2490 IFCH=1163THENN5$=N1$+N2$+N3$:RETURN
:rem 96
2500 GOTO2370 :rem 204
2510 POKE53281,1 :rem 87
2520 PRINT"{CLR}{ 2 SPAZI}{BLK}
{ 10 SPAZI}EROI DI CYLON ZAP":PRINT:
PRINT"{RED}{ 12 SPAZI}5 MIGLIORI PUN
TEGGI{OFF}" :rem 93
2530 PRINT"{HOME}{GIU'}{BLK}{ 4 GIU'}
{ 14 SPAZI}"A1$"...W1 :rem 127
2540 PRINT"{BLU}{ 2 GIU'}{ 14 SPAZI}"A2$
..."W2 :rem 203
2550 PRINT"{GRN}{ 2 GIU'}{ 14 SPAZI}"A3$
..."W3 :rem 205
2560 PRINT"{PUR}{ 2 GIU'}{ 14 SPAZI}"A4$
..."W4 :rem 78
2570 PRINT"{RED}{ 2 GIU'}{ 14 SPAZI}"A5$
..."W5 :rem 209
2580 RETURN :rem 175
2590 POKE54296,15:POKE54277,43:POKE54278,
73:POKE54276,129 :rem 226
2600 FORCO=127TO8STEP-17 :rem 196
2610 POKECS,CO :rem 76
2620 FORT=1TO100:NEXT:NEXTCO:POKECS,11
:rem 184
2630 IFS1=0THENSC=SC+1:GOSUB1890:POKEN1,3
2:N1=1042:S1=1 :rem 45
2640 IFS2=0THENSC=SC+1:GOSUB1890:POKEN2,3
2:N2=1922:S2=1 :rem 57
2650 IFS3=0THENSC=SC+1:GOSUB1890:POKEN3,3
2:N3=1464:S3=1 :rem 63
2660 IFS4=0THENSC=SC+1:GOSUB1890:POKEN4,3
2:N4=1502:S4=1 :rem 61
2670 FORS0=54272TO54272+28:POKES0,0:NEXT:
BOM=BOM-1:GOTO1260 :rem 68
2680 S=54272:FORE=STOS+28:POKEE,0:NEXT
:rem 104
2690 POKE54296,15:POKE54277,51:POKE54278,
84 :rem 171
2700 POKE54276,17:FORJ=1TO40STEP4:POKE542
73,J:POKE54272,255-J-25:NEXT
:rem 129
2710 FORT=1TO100:NEXT:POKE54276,32:FORT=1
TO50:NEXT :rem 134
2720 FORE=STOS+28:POKEE,0:NEXT:RETURN
:rem 175
2730 PRINT"{HOME}INSERIRE INIZIALI NUMERO
1":GOSUB1745:GOSUB2360:A1$=N5$:GOTO
2190 :rem 116
2740 PRINT"{HOME}INSERIRE INIZIALI NUMERO
2":GOSUB1745:GOSUB2360:A2$=N5$:GOTO
2190 :rem 119
2750 PRINT"{HOME}INSERIRE INIZIALI NUMERO
3":GOSUB1745:GOSUB2360:A3$=N5$:GOTO
2190 :rem 122
2760 PRINT"{HOME}INSERIRE INIZIALI NUMERO
4":GOSUB1745:GOSUB2360:A4$=N5$:GOTO
2190 :rem 125
2770 PRINT"{HOME}INSERIRE INIZIALI NUMERO
5":GOSUB1745:GOSUB2360:A5$=N5$:GOTO
2190 :rem 128

```

**Allegre,
Fresche,
Spiritose,
Pratiche.**

HOME COMPUTER
hce

LA RIVISTA DEL COMPUTER IN CASA
SETTEMBRE 1984 L. 3.500



**MUSICA, MUSICA, MUSICA: CHI
CON CHE COMPUTER, CON
ADAM, UN HOMENATO
ARRIVANO I NUOVI**

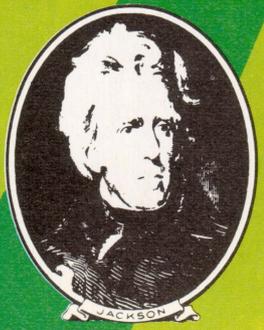
VIC 20, SEGA, SPECTRUM, TI 99/4A, I

**VIDEO
GIOCHI** 18
ISSN: 0392-89

LA PRIMA E UNICA RIVISTA ITALIANA DI VIDEOGAMES E GIOCOMPUTER
SETTEMBRE 1984 L. 3.500



SUPER CROSS FORCE
HIT BIT: I VOSTRI COMPUTER RECORD
TUTTO CHICAGO
INTERVISTA A DON BLUTH
IL CREATORE DI DRAGON'S LAIR
IL NUOVO ADAM



Con tutta la competenza del
**GRUPPO EDITORIALE
JACKSON**



Shuttle in fuga

di **Matt Giwer**
trad. ed adatt. di **U. Barzaghi**
e **R. Comini**

Siete in grado di manovrare attraverso un 'labirinto rotante' battendovi contro il tempo e evitando l'instancabile inseguimento dei Predoni? Impossibile? Provare per credere. Sono incluse versioni per il VIC 20 nella configurazione standard (con joystick) e per il C64 con joystick.

Durante una sosta per rifornirvi di carburante su di un pianeta fuori mano, incontrerete un misterioso vecchio che vi offre la chiave per un fantastico tesoro sepolto nelle viscere di una montagna. Molti hanno cercato di raggiungerlo ma tutti sono stati fermati dagli innominabili Predoni del Labirinto. Il vecchio ha il controllo principale di un labirinto rotante che sta a guardia del tesoro. In un ultimo anelito di vita mette nelle vostre mani un vecchio joystick Commodore. "Premi un pulsante", dice, "e controllerai il labirinto". Dal momento che non sapete resistere ad una avventura, partite alla volta della montagna.

Quando la raggiungete, vi trovate su di un lato di un labirinto in lento movimento. Di tanto in tanto il Predone rosso cupo vi attraversa il cammino. Se premete il pulsante del joystick, il labirinto aumenta la propria velocità; se lasciate il pulsante rallenterà. Armati di questo segreto entrate nel labirinto per raggiungere il tesoro verso l'estremità destra dello schermo.

Note alla versione per C64

La versione Commodore C64 di "labirinto rotante" è stata ribattezzata "Shuttle in fuga" poiché ha come soggetto una navetta spaziale.

"Shuttle in fuga" funziona suppergiù come le altre versioni. La differenza principale è data dall'aggiunta di una quantità chiamata GAS. Iniziate con 2000 unità di carburante che consumate ad un ritmo di 60 unità al secondo, sia che lo Shuttle si muova o no. Se toccate un muro o dei Predoni perdete 100 unità ogni sessantesimo di secondo. Quando avete esaurito il carburante, il gioco ha termine. Fortunatamente, potete rifornire i vostri serbatoi raggiungendo l'estremità destra dello schermo.

Se volete interrompere il gioco per un attimo, tenete premuto il tasto SHIFT. Se volete interromperlo più a lungo, usate SHIFT LOCK.

Potete aumentare la velocità di movimento dei muri tenendo premuto il pulsante di sparo del joystick. Questo non farà apparire i varchi più rapidamente, ma aumenterà la velocità di scorrimento dei varchi già presenti. Lo svantaggio consiste nel fatto che mentre il pulsante è premuto, il vostro carburante viene consumato a velocità doppia.

La programmazione di "Shuttle in fuga" ha messo in luce dei problemi interessanti. Il primo consiste nei "lampi" - piccole chiazze di neve - che appaiono sullo schermo. Normalmente questo

non provoca problemi, ma se cercate di usare il registro di controllo delle collisioni fra SPRITE e sfondo del chip VIC-II, vi accorgete che gli SPRITE collidono con i "lampi"!

Il significato di tutto ciò consiste nel fatto che occasionalmente, senza alcuna ragione apparente, lo Shuttle colliderà e voi perderete 100 unità di carburante. Dal momento che spostando il set di caratteri si eliminano i lampi, il set suddetto è stato ricollocato in \$3000.

Un altro cavillo del C64 è rappresentato dal fatto che il chip VIC-II può guardare solo 16K di memoria contemporaneamente. Quando accendete il vostro calcolatore, il chip vede il primo blocco di 16K dalla locazione \$0000 alla locazione \$3FFF. È stato deciso di lasciare le cose in questo stato, per semplicità. Questo significa che i dati degli SPRITE, il set di caratteri ricollocato, e l'intero programma BASIC devono essere concentrati in 16K.

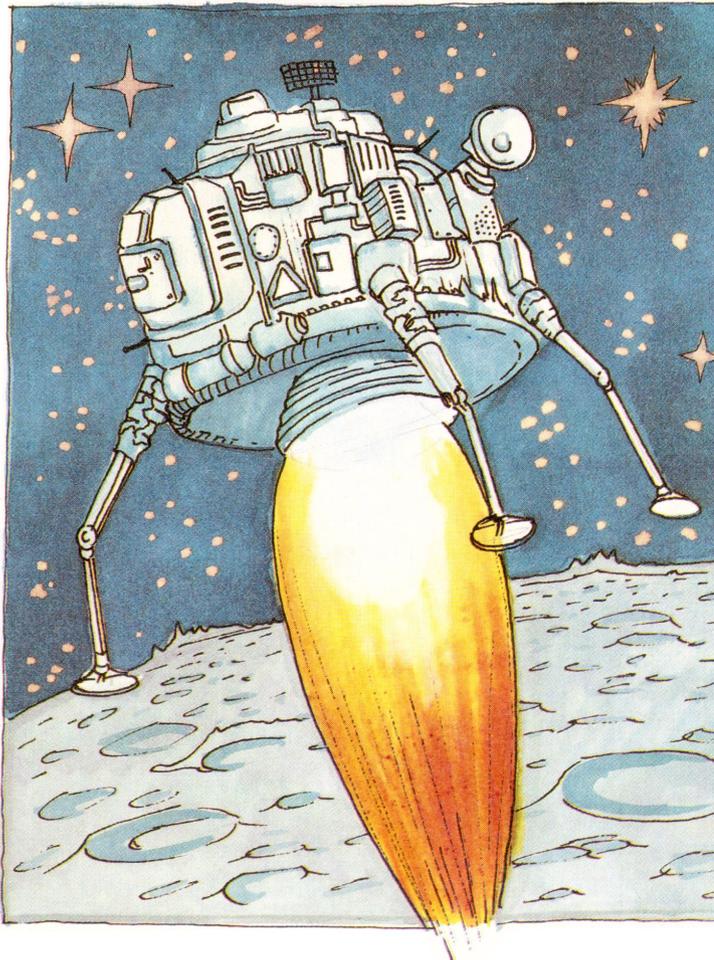
A causa di questa limitazione nell'occupazione di memoria, quando il linguaggio macchina crea un set di caratteri alla locazione \$3000, distrugge le istruzioni DATA nel programma. Fortunatamente, le istruzioni DATA non sono ulteriormente richieste, essendo già state inserite in memoria tramite istruzioni POKE.

Dal momento che il fatto di mandare il programma in esecuzione ne distrugge una parte, assicuratevi di averne registrato una copia prima di dare il RUN.

Shuttle in fuga versione per il VIC 20

```
10 POKE55,94:POKE56,29:CLR          :rem 25
15 PRINT"{HOME}"CHR$(8);:SS=36879:CS=3686
   5:FF=255:POKECS,FF:F8=484:Z4=4:Z5=5:RL
   =62:RR=63                          :rem 182
20 DS=36869:SE=8164:T3=23:F4=44:H4=64:V1=
   36875:DEFFNA(J)=Z3+J*Z3+S+F8*(JANDZ1)
                                       :rem 5
25 DIMT(5),D(4),K(4):Z3=3:SP=160:KS=197:Z
```

```
Q=2:RB=7673:RC=7665:RM=1:BU=198:S2=.94
   :SN=1                                :rem 155
30 H8=128:H3=32:S=7680:RS=57:Z0=0:Z1=1:SK
   =5:T2=22:T1=21:SH=57:ML=7518:ZB=176:LI
   =3                                    :rem 250
35 FORI=1TO4:READD(I),K(I):NEXT:JD=37154:
   J1=JD-3:J2=J1+1:POKEJD,127         :rem 11
40 FORI=1TO7:FORJ=0TO7:READA:POKE7616+I*8
   +J,A:NEXTJ,I                        :rem 82
45 FORI=7518TO7614:READA:POKEI,A:NEXT
                                       :rem 79
```



```

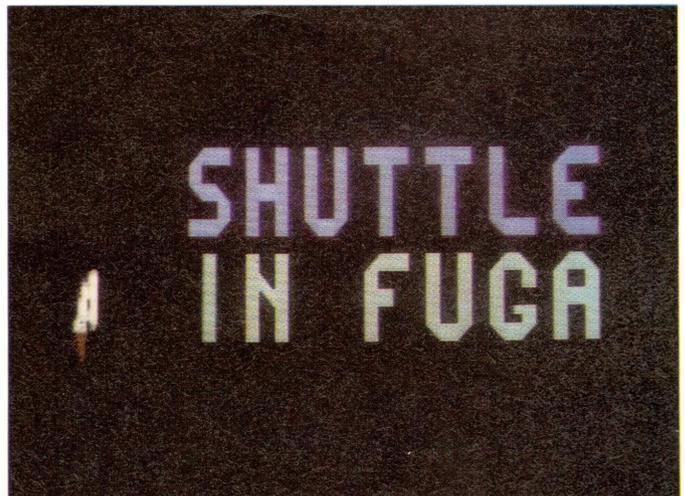
50 POKESS-1,10:A$="{RVS}{WHT}{ 3 SPAZI }
   {OFF}{RED}9{RVS}{ 2 SPAZI }{OFF}{CYN}9
   {RVS}{ 2 SPAZI }{OFF}{PUR}9{RVS}
   { 2 SPAZI }{OFF}{GRN}9{RVS}{ 2 SPAZI }
   {OFF}{BLU}9{RVS}{ 2 SPAZI }{OFF}{YEL}9
   {RVS} ":PRINT"{HOME}";:FORI=0TO21
                                     :rem 242
55 PRINTA$"{RVS}"CHR$(160) " ";:NEXT:PRINT
   A$"{RVS}"CHR$(160) "{HOME}";:POKE8185,1
   60                                     :rem 146
60 FORI=0TOLI-1:POKES+T1+T2*I,61:NEXT:POK
   ESS,8:POKEDS,FF:POKECS,25:GOTO265
                                     :rem 2
65 POKEV1,240:OE=OE-Z1                                     :rem 58
70 FORJ=0TO5:IFT(J)=Z0THEN85                                     :rem 248
75 T(J)=T(J)+Z1:IFT(J)>SKTHENPOKEFNA(J),R
   S:T(J)=Z0                                     :rem 96
80 GOTO90                                                 :rem 10
85 IFRND(Z1)>S2THENT(J)=Z1:POKEFNA(J),SP
   90                                     :rem 13
90 NEXT:POKEV1,Z0:POKESL,E1:SYSML:IFPEEK(
   SL)=E1ORPEEK(SL)=SPTHENPOKESL,OS
   95                                     :rem 188
95 RETURN                                                 :rem 78
100 IFROTHEN110                                           :rem 130
105 R1=RND(Z1)>.5:RD=T2+F4*R1:RT=INT(RND(
   Z1)*Z5)*Z3+Z4-F8*R1+S                                     :rem 180
110 POKEV1,200:POKERC+RM,FF:POKERB+RM,FF:
   IFR1THEN125                                             :rem 24
115 RM=RM+Z1:IFRM>Z4THENRM=Z0                             :rem 145
120 GOTO130                                               :rem 96
125 RM=RM-Z1:IFRM<Z1THENRM=Z5                             :rem 148
130 POKERC+RM,H8:POKERB+RM,Z1                             :rem 76
135 POKERT,SP:POKERT+Z1,SP:RT=RT+RD:RO=RO
   +Z1:IFRO=T3THENRO=Z0:POKEV1,Z0:RETURN

```

```

                                     :rem 73
140 POKERT,RL:POKERT+Z1,RR:POKEV1,Z0:RETU
   RN                                                     :rem 137
145 IFPEEK(SL)<>OSTHENEM=PEEK(SL):GOTO240
   150                                     :rem 118
150 IF(PEEK(KS)=H3)=Z0AND(PEEK(J1)ANDH3)T
   HEN165                                                 :rem 55
155 'ZW=(ZW+Z1)ANDZ3:IFZW=Z0THENZW=Z1
   160                                     :rem 138
160 ONZWGOSUB65,100,65:GOTO145                             :rem 49
165 DI=Z0:A=PEEK(KS):IFA<>H4THEN195
   170                                     :rem 138
170 IF(PEEK(J2)ANDH8)=Z0THENDI=Z4:GOTO205
   175                                     :rem 21
175 J=PEEK(J1):IF(JANDZ4)=Z0THENDI=Z1:GOT
   O205                                                   :rem 47
180 IF(JAND8)=Z0THENDI=2:GOTO205:rem 202
185 IF(JAND16)=Z0THENDI=Z3                               :rem 79
190 GOTO205                                               :rem 106
195 FORI=1TO4:IFA=K(I)THENDI=I                           :rem 130
200 NEXT                                                 :rem 209
205 IFDI=Z0THEN235                                       :rem 67
210 IFDI=Z3THENOE=OE-SN:IFINT((SL-S)/T2)=
   (SL-S)/T2THEN235                                     :rem 48
215 IFDI=Z4THENOE=OE+SN                                  :rem 223
220 EL=SL+D(DI):IFEL<SOREL>SE+T1THEN235
   225                                     :rem 184
225 POKEV1,220:E1=EM:EM=PEEK(EL):IFEM<>SP
   THEN240                                               :rem 219
230 POKESL,SP:OS=SH+DI:POKEEL,OS:SL=EL
   235                                     :rem 13
235 ZQ=Z3-ZQ:POKEV1,Z0:ONZQGOSUB70,100:GO
   TO145                                                 :rem 192
240 IFEM<>224THEN255                                       :rem 148
245 POKEV1,0:SN=SN+1:SK=SK-1:IFSK<2THENSK
   =2                                                     :rem 228
250 FORI=1TO5:POKESS,25:A=TAN(I):POKESS,8
   :A=TAN(I):NEXT:GOTO265                               :rem 21
255 POKESL,RS:POKEV1,0:POKEV1+2,129:FORI=
   15TO0STEP-1:POKESS-1,I:A=TAN(I):NEXT
   260                                     :rem 166
260 POKEV1+2,0:POKESS-1,10:LI=LI-1:IFLI=-
   1THEN270                                             :rem 193
265 POKESL,E1:SL=7922:OS=61:EM=OS:E1=EM:P
   OKESL,OS:POKES+T1+T2*LI,SP:GOTO145
   270                                     :rem 47
270 POKEV1,0:POKECS,FF:POKESS,42:POKEDS,2
   40:PRINT"{CLR}{OFF}{ 2 GIU' }{WHT} HAI
   USATO TUTTE LE{ 2 SPAZI}"                             :rem 161
275 PRINTSPC(7)"ASTRONAVI...{GIU'}"
   47

```



```

280 PRINTSPC(7)"PUNTI:"OE:IFOE>HITENHI=0
E:PRINT"{RVS}"; :rem 76
285 PRINT"{ 2 DES}{ 2 SPAZI}RECORD
{ 2 SPAZI}{OFF}:"HI:OE=0 :rem 1
290 PRINT"{GIU'}{ 2 SPAZI}GIOCHI ANCORA(S
/N)?" :POKECS,25:POKEBU,0 :rem 50
295 GETA$:IFA$<>"S"ANDA$<>"N"THEN295
:rem 55
300 IFA$="S"THENPOKECS,255:LI=3:SK=5:GOTO
50 :rem 120
305 PRINT"{CLR}{BLU}"; :POKEJD,FF:POKESS,2
7:POKECS,38:END :rem 80
310 DATA-22,12,22,36,-1,20,1,44 :rem 252
315 DATA126,126,126,126,126,126,126,126
:rem 175
320 DATA16,56,108,198,130,254,146,130
:rem 78
325 DATA130,146,254,130,198,108,56,16
:rem 83
330 DATA31,52,100,198,100,52,31,0
:rem 110
335 DATA248,44,38,99,38,44,248,0:rem 109
340 DATA127,255,255,255,255,255,255,127
:rem 193
345 DATA254,255,255,255,255,255,255,254
:rem 200
350 DATA162,3,134,0,160,3,162,22,169,30,1
33,2,169,0,133,1 :rem 5
355 DATA177,1,133,251,24,165,1,105,22,133
,1,144,2,230,2,177 :rem 112
360 DATA1,133,252,165,251,145,1,165,252,1
33,251,202,208,230,200,200 :rem 242
365 DATA200,162,22,169,31,133,2,169,228,1
33,1,177,1,133,251,56 :rem 22
370 DATA165,1,233,22,133,1,176,2,198,2,17
7,1,133,252,165,251 :rem 177
375 DATA145,1,165,252,133,251,202,208,230
,200,200,200,198,0,208,166,96
:rem 153

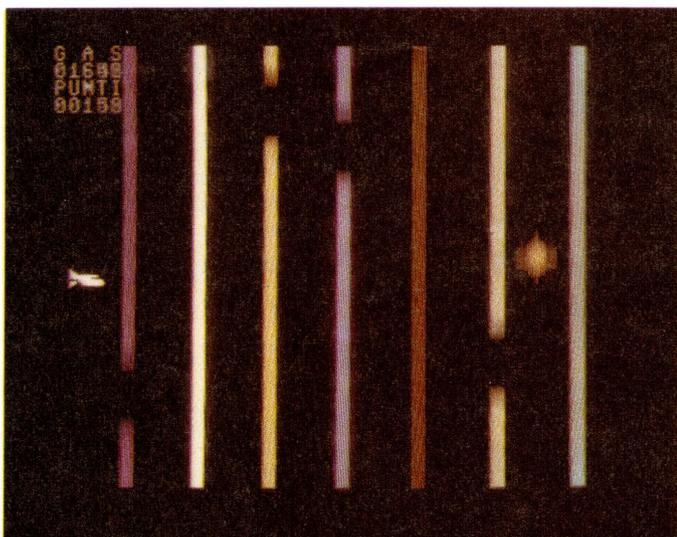
```

Shuttle in fuga versione per il C64

```

100 GOSUB3000 :rem 212
110 PRINTCHR$(142) :rem 8
120 IF PEEK(16378)<>16 THEN GOSUB 10000:G
OSUB 50000 :rem 209

```



```

130 V=13*4096:CO=50 :rem 246
140 POKE V+16,0 :rem 8
150 POKE V+39,1:POKEV+40,1 :rem 186
160 POKE V+0,CO:POKEV+4,CO-2:POKEV+5,221
:rem 80
170 POKE V+1,200 :rem 55
180 POKE V+2,CO :rem 57
190 POKE V+3,179 :rem 74
200 POKE V+21,3 :rem 4
210 POKE V+39,1:POKEV+40,1:POKEV+41,8:POK
EV+42,1 :rem 23
220 POKE 2040,245 :rem 80
230 POKE 2041,246:POKE2043,246 :rem 77
240 POKE 2042,247 :rem 86
250 FOR K=1 TO 500 : NEXT K:POKEV+21,7
:rem 227
260 GOSUB 2000 :rem 218
270 I=200 :rem 177
280 P=1 :rem 88
290 Q=Q+.01*P :rem 174
300 P=P+.1:C=C+1 :rem 83
310 I=I-Q :rem 225
320 IF PEEK(2042)=248 THEN POKE 2042,254:
GOTO340 :rem 52
330 IF PEEK(2042)=254 THEN POKE 2042,248
:rem 43
340 POKE V+1,I:POKEV+3,I-21:POKEV+5,I+21
:rem 49
350 POKES+22,P2:POKES+23,1OR(16-P2/16)*16
:rem 47
360 P2=P2+P2/244 :rem 80
370 IF C=60THEN POKEV+23,4 :rem 179
380 IF C=20THEN POKEV+23,4 :rem 176
390 IF C=40 THEN POKEV+23,0:POKE2042,248
:rem 170
400 IF C<70 THEN 290 :rem 211
410 POKE 2040,249 :rem 85
420 POKE 2043,250:POKEV+6,CO:POKEV+7,I:PO
KEV+21,15 :rem 164
430 Q=Q+.01*P :rem 170
440 P=P+.1:C=C+1:C2=C2+.6 :rem 72
450 I=I-Q :rem 230
460 IF PEEK(2042)=248 THEN POKE 2042,254:
GOTO480 :rem 62
470 IF PEEK(2042)=254 THEN POKE 2042,248
:rem 48
480 POKE V+1,I:POKEV+5,I+21 :rem 20
490 NR=I+C2*C2:NC=CO+C2*3 :rem 153
500 POKE V+7,NR:POKEV+3,NR-21:POKEV+6,NC:
POKEV+2,NC :rem 119
510 IF C=83 THEN POKE 2043,251:POKE2041,2
53 :rem 241
520 IF C=86 THEN POKE 2043,252 :rem 0
530 IF C=89 THEN POKE V+21,5 :rem 187
540 POKES+22,P2:POKES+23,1OR(16-P2/16)*16
:rem 48
550 P2=P2+P2/244 :rem 81
560 IF I>25 THEN 430 :rem 222
570 POKE S+4,128 :rem 68
580 POKE V+5,I+21 :rem 133
590 I=I-2:IFI>0 THEN580 :rem 24
600 POKE V+21,1 :rem 6
610 FOR J=1 TO 2000:NEXT :rem 20
620 PRINT"{HOME}{ 10 DES}{WHT}{ 2 SPAZI}
ORBITA RAGGIUNTA..." :rem 144
630 FOR I=1 TO 1000:NEXT :rem 20
640 POKE 2040,244 :rem 85
650 POKE V,0:POKEV+1,117 :rem 92
660 FOR I=0 TO 348 STEP2 :rem 233
670 POKE V,I AND 255:POKEV+16,I/255
:rem 151

```

```

680 NEXT :rem 221
690 FOR I=0 TO 1000:NEXT :rem 25
700 GOTO 4000 :rem 148
710 POKE 53281,12:POKE53280,12:PRINT"
  {WHT} :rem 65
720 END :rem 112
2000 S=54272 :rem 86
2010 POKES+24,15+16+32:POKES+23,1+16*5
      :rem 36
2020 POKES+5,0 :rem 2
2030 POKES+6,16*15+15 :rem 92
2040 POKES+4,129 :rem 111
2050 POKES+1,11 :rem 51
2060 P2=100:RETURN :rem 50
3000 POKE 53281,0:POKE53280,0 :rem 24
3010 PRINT"{CLR}" :rem 40
3020 PRINT"{ 5 GIU' }" :rem 235
3040 T=12 :rem 187
3050 PRINTTAB(T)" [<*>]{RVS}ε{ 2 SPAZI}
  {DES} {DES} {DES} {DES} {DES}
  { 3 SPAZI}{DES}{ 3 SPAZI}{DES}
  { 3 DES}ε{ 2 SPAZI}" :rem 53
3060 PRINTTAB(T)" {RVS} { 3 DES} {DES}
  {DES} {DES} { 2 DES} { 3 DES}
  { 2 DES} { 3 DES} " :rem 248
3070 PRINTTAB(T)" [<*>]{RVS} [<*>]{DES}
  { 3 SPAZI}{DES} {DES} { 2 DES}
  { 3 DES} { 2 DES} { 3 DES}
  { 2 SPAZI}" :rem 96
3080 PRINTTAB(T)" {RVS}{ 2 DES} {DES}
  {DES} {DES} {DES} { 2 DES} { 3 DES}
  { 2 DES} { 3 DES} " :rem 250
3090 PRINTTAB(T)" {RVS}{ 2 SPAZI}{OFF}ε
  {RVS}{DES} {DES} {DES}{OFF}[<*>]
  {RVS} {OFF}ε{RVS}{ 2 DES} { 3 DES}
  { 2 DES}{OFF}[<*>]{RVS}{ 2 SPAZI}
  {DES}{OFF}[<*>]{RVS}{ 2 SPAZI}"
      :rem 141
3100 PRINT :rem 81
3110 PRINTTAB(T)" {CYN} {RVS} {OFF}
  { 2 SPAZI}{RVS} {OFF} {RVS} {OFF}
  { 5 SPAZI}{RVS}ε{ 2 SPAZI}{OFF}
  {RVS} {OFF} {RVS} {OFF} {RVS}ε [<*>]
  {OFF} {RVS}ε [<*>]{OFF}" :rem 138
3120 PRINTTAB(T)" {RVS} {OFF}{ 2 SPAZI}
  {RVS} [<*>] {OFF}{ 5 SPAZI}{RVS}
  {OFF}{ 3 SPAZI}{RVS} {OFF} {RVS}
  {OFF} {RVS} {OFF} {RVS} {OFF} {RVS}
  {OFF} {RVS} {OFF}" :rem 182
3130 PRINTTAB(T)" {RVS} {OFF}{ 2 SPAZI}
  {RVS} {OFF}[<*>]{RVS} {OFF}
  { 5 SPAZI}{RVS}{ 2 SPAZI}{OFF}
  { 2 SPAZI}{RVS} {OFF} {RVS} {OFF}
  {RVS} {OFF}{ 3 SPAZI}{RVS}
  { 3 SPAZI}{OFF}" :rem 19
3140 PRINTTAB(T)" {RVS} {OFF}{ 2 SPAZI}
  {RVS} {OFF} {RVS} {OFF}{ 5 SPAZI}
  {RVS} {OFF}{ 3 SPAZI}{RVS} {OFF}
  {RVS} {OFF} {RVS} {OFF}[<C>]{RVS}
  {OFF} {RVS} {OFF} {RVS} {OFF}"
      :rem 57
3150 PRINTTAB(T)" {RVS} {OFF}{ 2 SPAZI}
  {RVS} {OFF} {RVS} {OFF}{ 5 SPAZI}
  {RVS} {OFF}{ 3 SPAZI}[<*>]{RVS}
  {OFF}ε [<*>]{RVS} {OFF}ε {RVS} {OFF}
  {RVS} {OFF}" :rem 70
3999 RETURN :rem 190
4000 V=13*4096:POKE53281,0:POKE53280,0
      :rem 71
4010 POKE V+21,0:POKEV+23,0 :rem 223

```

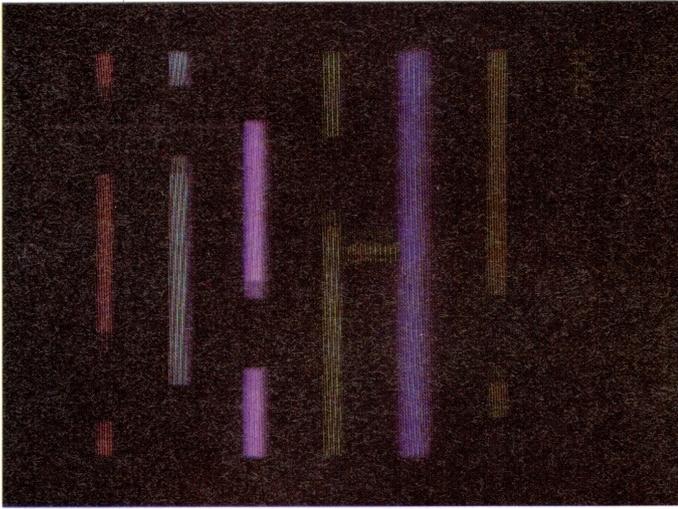




```

4020 GOSUB 2000:POKE S+5,7*16:POKES+6,249
      :POKES+4,128 :rem 226
4030 POKE V+40,03:POKEV+41,07:POKEV+42,03
      :POKEV+43,07:POKEV+44,03:POKEV+45,07
      :rem 214
4040 POKE V+46,03:FOR I=1 TO 6 : POKE V+2
      *I,(36+40*I)AND255:NEXT :rem 129
4050 POKE V+16,64:POKE 2040,244:POKEV,30:
      POKEV+1,148:POKE V+21,255 :rem 178
4060 FOR I=2041 TO 2047:POKEI,255:NEXT
      :rem 97
4070 PRINT"{CYN}{CLR}G A S :rem 135
4080 PRINT"02000" :rem 143
4090 PRINT"PUNTI:" :rem 104
4100 PRINT"00000" :rem 134
4110 P(0)=1029:P(4)=1994:P(1)=1039:P(5)=2
      004:P(2)=1049:P(6)=2014:P(3)=1059
      :rem 51
4120 SYS 49152 :rem 203
4130 POKE P(0),227 :rem 143
4140 IF PEEK(2)=255 THEN 20000 :rem 250
4150 IF PEEK(653)=1 THEN 4150 :rem 212
4160 IF RND(1)>.05 THEN 4140 :rem 137
4170 IF RND(1)>.5 THEN 4200 :rem 87
4180 P=RND(1)*5:IF PEEK(P(P))<>160 THEN 4
      180 :rem 92
4190 POKE P(P),227:GOTO4140 :rem 241
4200 P=RND(1)*3+4:IF PEEK(P(P))<>160 THEN
      4200 :rem 171
4210 POKE P(P),228:GOTO4140 :rem 235
5000 FB=(J AND 16) :rem 95
5010 IFFL=0AND FB=0 THEN POKE 2,0:POKE 49
      290,2:GOTO 5030 :rem 18
5020 IF FL=16 AND FB=16 THEN POKE 2,0:POK
      E 49290,3 :rem 71
5030 FL=FB :rem 31
5040 IF (J AND 8)=0 AND DX<4 THEN DX=DX+1
      :GOTO5060 :rem 136
5050 IF (J AND 4)=0 AND DX>-4 THEN DX=DX-
      1 :rem 120
5060 IF (J AND 1)=0 AND DY>-4 THEN DY=DY-
      1:GOTO5080 :rem 185
5070 IF (J AND 2)=0 AND DY<4 THEN DY=DY+1
      :rem 74
5080 X=PEEK(V)+PEEK(V+16)*256 :rem 126
5085 Y=PEEK(V+1) :rem 144
5090 NX=X+DX:IF NX>21 AND NX<358 THEN POK
      E V,NXAND255:POKEV+16,NX/256
      :rem 25
5100 NY=Y+DY:IF NY<20 THEN NY=210
      :rem 127
5110 IF NY>210 THEN NY=20 :rem 67
5120 POKE V+1,NY :rem 124
5130 RETURN :rem 169
10000 I=15616:TI$="000000" :rem 22
10005 PRINT"{HOME}{WHT}{ 12 DES}PRONTO IN
      "LEFT$(STR$(140-INT(TI/60)),4)" SEC
      ONDI " :rem 155
10010 READ A:IF A=256 THEN RETURN
      :rem 66

```



```

10020 POKE I,A:I=I+1:GOTO 10005 :rem 166
10030 DATA 0,0,0,0,0,0,0 :rem 102
10040 DATA 0,0,24,0,0,28,0 :rem 215
10050 DATA 0,31,0,0,31,255,240 :rem 162
10060 DATA 31,255,8,20,255,254,31 :rem 78
10070 DATA 127,255,30,63,254,24,0 :rem 77
10080 DATA 0,0,0,0,0,0,0 :rem 107
10090 DATA 0,0,0,0,0,0,0 :rem 108
10100 DATA 0,0,0,0,0,0,0 :rem 100
10110 DATA 0,0,0,0,0,0,0 :rem 101
10120 DATA 0,0,71,192,0,247,192 :rem 227
10130 DATA 0,247,192,1,255,192,2 :rem 27
10140 DATA 255,192,2,255,192,2,247 :rem 137
10150 DATA 192,2,247,192,3,247,192 :rem 140
10160 DATA 3,247,192,3,247,192,3 :rem 37
10170 DATA 247,192,3,247,192,3,247 :rem 144
10180 DATA 192,3,255,192,3,255,192 :rem 142
10190 DATA 7,103,192,7,103,192,15 :rem 81
10200 DATA 229,128,31,119,128,31,240 :rem 225
10210 DATA 0,0,0,0,0,0,0 :rem 102
10220 DATA 0,0,0,0,0,0,0 :rem 103
10230 DATA 0,0,0,0,0,0,0 :rem 104
10240 DATA 0,0,0,0,0,0,0 :rem 105
10250 DATA 0,0,0,0,0,0,0 :rem 106
10260 DATA 0,0,0,0,0,0,0 :rem 107
10270 DATA 0,0,0,0,0,0,0 :rem 108
10280 DATA 0,0,3,128,0,15,192 :rem 125
10290 DATA 0,15,192,0,15,192,0 :rem 178
10300 DATA 15,192,0,1,252,0,1 :rem 115
10310 DATA 116,0,1,212,0,0,88 :rem 117
10320 DATA 0,0,80,0,0,0,0 :rem 160
10330 DATA 0,0,0,0,0,0,0 :rem 105
10340 DATA 0,0,0,0,0,0,0 :rem 106
10350 DATA 0,0,0,0,0,0,0 :rem 107
10360 DATA 0,0,0,0,0,0,0 :rem 108
10370 DATA 0,0,0,0,0,0,0 :rem 109
10380 DATA 0,0,0,0,0,0,0 :rem 110
10390 DATA 0,0,0,0,1,252,0 :rem 217
10400 DATA 1,252,0,1,252,0,1 :rem 60
10410 DATA 254,0,7,248,0,6,249 :rem 189
10420 DATA 0,2,251,0,6,122,0 :rem 62
10430 DATA 3,242,0,0,248,0,0 :rem 67
10440 DATA 248,0,0,60,0,0,120 :rem 114

```

```

10450 DATA 0,0,56,0,0,56,0 :rem 226
10460 DATA 0,96,0,0,96,0,0 :rem 235
10470 DATA 8,0,0,32,0,0,0 :rem 171
10480 DATA 0,0,0,0,0,0,64 :rem 169
10490 DATA 0,0,240,0,0,240,0 :rem 60
10500 DATA 1,240,0,2,240,0,2 :rem 57
10510 DATA 240,0,2,240,0,2,240 :rem 159
10520 DATA 0,3,240,0,3,240,0 :rem 60
10530 DATA 3,240,0,3,240,0,3 :rem 64
10540 DATA 240,0,3,240,0,3,240 :rem 164
10550 DATA 0,3,240,0,7,96,0 :rem 28
10560 DATA 7,96,0,15,224,0,31 :rem 134
10570 DATA 112,0,31,240,0,0,0 :rem 109
10580 DATA 7,192,0,7,192,0,7 :rem 93
10590 DATA 192,0,7,192,0,7,192 :rem 195
10600 DATA 0,7,192,0,7,192,0 :rem 79
10610 DATA 7,192,0,7,192,0,7 :rem 87
10620 DATA 192,0,7,192,0,7,192 :rem 189
10630 DATA 0,7,192,0,7,192,0 :rem 82
10640 DATA 7,192,0,7,192,0,7 :rem 90
10650 DATA 192,0,7,192,0,7,192 :rem 192
10660 DATA 0,3,128,0,0,0,0 :rem 221
10670 DATA 0,2,0,0,7,192,0 :rem 229
10680 DATA 7,192,0,6,192,0,4 :rem 90
10690 DATA 192,0,3,64,0,6,192 :rem 141
10700 DATA 0,1,192,0,4,0,0 :rem 219
10710 DATA 7,192,0,7,128,0,7 :rem 87
10720 DATA 64,0,7,192,0,1,192 :rem 134
10730 DATA 0,5,192,0,6,64,0 :rem 30
10740 DATA 7,192,0,7,192,0,0 :rem 84
10750 DATA 128,0,3,128,0,0,0 :rem 72
10760 DATA 0,0,2,0,0,1,0 :rem 115
10770 DATA 0,6,64,0,0,64,0 :rem 235
10780 DATA 4,128,0,3,64,0,6 :rem 36
10790 DATA 0,0,1,0,0,0,0 :rem 116
10800 DATA 0,0,0,0,0,128,0 :rem 214
10810 DATA 1,64,0,6,0,0,1 :rem 174
10820 DATA 0,0,5,0,0,6,64 :rem 178
10830 DATA 0,0,0,0,4,0,0 :rem 114
10840 DATA 0,128,0,3,128,0,0 :rem 72
10850 DATA 0,0,0,0,0,0,0 :rem 112
10860 DATA 0,0,0,0,0,0,0 :rem 113
10870 DATA 0,0,0,0,0,0,0 :rem 114
10880 DATA 0,0,0,0,0,0,0 :rem 115
10890 DATA 0,0,0,0,0,0,0 :rem 116
10900 DATA 0,0,0,0,0,0,0 :rem 108
10910 DATA 0,0,0,0,0,0,0 :rem 109
10920 DATA 0,0,1,128,0,6,128 :rem 75
10930 DATA 0,2,64,0,5,192,0 :rem 28
10940 DATA 3,128,0,1,252,0,1 :rem 73
10950 DATA 252,0,1,236,0,1,126 :rem 176
10960 DATA 0,3,248,0,2,120,0 :rem 72
10970 DATA 0,248,0,0,120,0,0 :rem 68
10980 DATA 112,0,0,120,0,0,120 :rem 158
10990 DATA 0,0,48,0,0,48,0 :rem 237
11000 DATA 0,0,0,0,0,0,0 :rem 100
11010 DATA 0,0,0,0,0,0,0 :rem 101
11020 DATA 0,0,0,0,0,0,0 :rem 102
11030 DATA 0,0,0,0,0,0,0 :rem 103
11040 DATA 0,16,0,0,16,0,16 :rem 13
11050 DATA 56,16,10,16,160,4,16 :rem 229
11060 DATA 64,10,124,160,1,255,0 :rem 16
11070 DATA 1,255,0,11,255,144,127 :rem 73
11080 DATA 255,252,11,255,144,1,255 :rem 181
11090 DATA 0,1,255,0,10,124,160 :rem 217
11100 DATA 4,16,64,10,16,160,16 :rem 224
11110 DATA 56,16,0,16,0,0,16 :rem 70
11120 DATA 0,0,0,0,0,0,256 :rem 212
20000 SC=0:FOR I=0 TO 4:SC=SC+(PEEK(1148-

```



**Shuttle
in fuga**

```

I)-48)*10↑I:NEXT I           :rem 9
20010 IF H<SC THEN H=SC       :rem 230
20020 POKE S+4,128            :rem 156
20030 POKE 13*4096+21,0      :rem 111
20040 FOR I=1 TO 1000:NEXT I  :rem 186
20050 PRINT"{CLR}GAS TERMINATO...{GIU'}
                               :rem 98
20060 PRINT"HAI TOTALIZZATO {WHT}"SC"
      {CYN}PUNTI              :rem 236
20070 PRINT"RECORD{WHT}"H"{CYN} :rem 151
20080 PRINT"{ 3 GIU'}{ 11 SPAZI}ANCORA? (
      S/N)"                   :rem 18
20090 PRINT"{GIU'} OPPURE PREMI IL PULSAN
      TE PER RIPARTIRE        :rem 173
20100 GETA$                   :rem 56
20110 IF A$="N"THEN END       :rem 189
20120 IF (PEEK(56320) AND 16)=0 THEN GOTO
      4000                    :rem 30
20130 IF A$<>"Y" THEN 20100   :rem 35
20140 GOTO4000                :rem 244
50000 I=49152:TI$="000000"    :rem 28
50010 PRINT"{HOME}{WHT}{ 12 DES}PRONTO IN
      "LEFT$(STR$(97-INT(TI/60)),4)" SECO
      NDI "                   :rem 118
50015 READ A:IF A=256 THEN PRINT"{HOME}
      { 10 DES}{ 21 SPAZI}{SPAZI}":RETURN
                               :rem 43
50020 IF A=-1 THEN I=49920 : GOTO 50010
                               :rem 136
50030 IF A=-2 THEN I=50688 : GOTO 50010
                               :rem 141
50040 POKE I,A:I=I+1:GOTO 50010 :rem 172
50050 DATA 120,169,0,141,20,3,169
                               :rem 74
50060 DATA 195,141,21,3,88,173,14
                               :rem 88
50070 DATA 220,41,254,141,14,220,165
                               :rem 221
50080 DATA 1,41,251,133,1,160,0 :rem 220
50090 DATA 185,0,208,153,0,48,185
                               :rem 91
50100 DATA 0,50,153,0,50,185,0 :rem 169
50110 DATA 209,153,0,49,185,0,211
                               :rem 76
50120 DATA 153,0,51,185,0,212,153
                               :rem 69
50130 DATA 0,52,185,0,213,153,0 :rem 223
50140 DATA 53,185,0,214,153,0,54 :rem 27
50150 DATA 185,0,215,153,0,55,169
                               :rem 86
50160 DATA 15,141,156,200,200,208,200
                               :rem 6
50170 DATA 165,1,9,4,133,1,173 :rem 188
50180 DATA 14,220,9,1,141,14,220 :rem 18
50190 DATA 169,28,141,24,208,169,15
                               :rem 199
50200 DATA 141,156,200,169,255,141,15
                               :rem 21
50210 DATA 212,169,128,141,18,212,169
                               :rem 30
50220 DATA 0,133,2,141,224,207,141
                               :rem 113
50230 DATA 255,207,141,254,207,141,253
                               :rem 75
50240 DATA 207,141,252,207,141,249,207
                               :rem 76
50250 DATA 160,6,169,20,153,0,207
                               :rem 79
50260 DATA 169,0,153,16,207,136,208
                               :rem 188
50270 DATA 243,169,251,141,251,207,160

```

```

                               :rem 79
50280 DATA 0,169,4,133,252,132,251
                               :rem 131
50290 DATA 169,216,133,254,132,253,169
                               :rem 93
50300 DATA 160,160,5,145,251,160,10
                               :rem 167
50310 DATA 145,251,160,15,145,251,160
                               :rem 19
50320 DATA 20,145,251,160,25,145,251
                               :rem 224
50330 DATA 160,30,145,251,160,35,145
                               :rem 226
50340 DATA 251,165,251,24,105,40,133
                               :rem 225
50350 DATA 251,144,2,230,252,201,232
                               :rem 218
50360 DATA 208,211,169,1,160,10,145
                               :rem 177
50370 DATA 253,169,4,160,5,145,253
                               :rem 143
50380 DATA 169,7,160,15,145,253,169
                               :rem 202
50390 DATA 14,160,20,145,253,169,8
                               :rem 141
50400 DATA 160,25,145,253,169,13,160
                               :rem 232
50410 DATA 30,145,253,169,3,160,35
                               :rem 133
50420 DATA 145,253,165,253,24,105,40
                               :rem 231
50430 DATA 133,253,144,2,230,254,201
                               :rem 221
50440 DATA 232,208,199,96,-1 :rem 104
50450 DATA 173,141           :rem 117
50460 DATA 2,201,1,208,3,76,49 :rem 190
50470 DATA 234,230,2,165,2,201,2 :rem 21
50480 DATA 240,3,76,49,234,169,0 :rem 47
50490 DATA 133,2,169,3,133,252,169
                               :rem 144
50500 DATA 216,133,251,160,45,177,251
                               :rem 27
50510 DATA 32,79,195,160,55,177,251
                               :rem 201
50520 DATA 32,79,195,160,65,177,251
                               :rem 203
50530 DATA 32,79,195,160,75,177,251
                               :rem 205
50540 DATA 32,79,195,165,251,24,105
                               :rem 196
50550 DATA 40,133,251,144,2,230,252
                               :rem 173
50560 DATA 201,192,208,213,76,0,198
                               :rem 192
50570 DATA 201,160,240,19,201,32,240
                               :rem 219
50580 DATA 37,162,1,232,221,174,195
                               :rem 191
50590 DATA 208,250,202,189,174,195,145
                               :rem 97

```

50600 DATA 251,96,152,56,233,40,168		50960 DATA 100,32,173,0,220,72,41	
:rem 194		:rem 73	
50610 DATA 177,251,201,32,240,1,96		50970 DATA 15,201,15,240,8,169,129	
:rem 131		:rem 144	
50620 DATA 152,24,105,40,168,169,227		50980 DATA 141,4,212,76,183,198,169	
:rem 241		:rem 210	
50630 DATA 145,251,96,165,252,201,3		50990 DATA 128,141,4,212,104,41,16	
:rem 188		:rem 132	
50640 DATA 240,22,152,56,233,40,168		51000 DATA 205,255,207,240,48,141,255	
:rem 185		:rem 24	
50650 DATA 177,251,201,160,240,1,96		51010 DATA 207,201,16,208,24,169,2	
:rem 185		:rem 126	
50660 DATA 152,24,105,40,168,169,99		51020 DATA 141,15,195,169,1,141,252	
:rem 204		:rem 181	
50670 DATA 145,251,96,152,24,105,120		51030 DATA 198,141,229,200,169,0,141	
:rem 236		:rem 232	
50680 DATA 168,177,251,201,100,240,1		51040 DATA 250,207,141,224,207,76,239	
:rem 230		:rem 30	
50690 DATA 96,152,56,233,120,168,169		51050 DATA 198,169,1,141,15,195,169	
:rem 2		:rem 203	
50700 DATA 99,145,251,96,160,228,239		51060 DATA 2,141,252,198,141,229,200	
:rem 2		:rem 230	
50710 DATA 249,226,120,119,99,32,32		51070 DATA 169,0,133,2,32,245,198	
:rem 194		:rem 90	
50720 DATA 100,111,121,98,248,247,227		51080 DATA 76,32,200,238,250,207,173	
:rem 31		:rem 236	
50730 DATA -2,169,7,133,252		51090 DATA 250,207,201,1,240,1,96	
:rem 47		:rem 75	
50740 DATA 169,32,133,251,160,170,177		51100 DATA 169,0,141,250,207,173,0	
:rem 36		:rem 122	
50750 DATA 251,32,47,198,160,180,177		51110 DATA 220,141,254,207,41,1,208	
:rem 252		:rem 168	
50760 DATA 251,32,47,198,160,190,177		51120 DATA 13,173,253,207,201,253,240	
:rem 254		:rem 16	
50770 DATA 251,32,47,198,165,251,56		51130 DATA 23,206,253,207,76,45,199	
:rem 206		:rem 197	
50780 DATA 233,40,133,251,176,2,198		51140 DATA 173,254,207,41,2,208,10	
:rem 195		:rem 126	
50790 DATA 252,201,56,208,220,76,160		51150 DATA 173,253,207,201,3,240,3	
:rem 240		:rem 123	
50800 DATA 198,201,160,240,19,201,32		51160 DATA 238,253,207,173,254,207,41	
:rem 227		:rem 35	
50810 DATA 240,37,162,1,232,221,142		51170 DATA 8,208,13,173,252,207,201	
:rem 173		:rem 182	
50820 DATA 198,208,250,202,189,142,198		51180 DATA 3,240,23,238,252,207,76	
:rem 99		:rem 139	
50830 DATA 145,251,96,152,24,105,40		51190 DATA 82,199,173,254,207,41,4	
:rem 187		:rem 151	
50840 DATA 168,177,251,201,32,240,1		51200 DATA 208,10,173,252,207,201,253	
:rem 184		:rem 15	
50850 DATA 96,152,56,233,40,168,169		51210 DATA 240,3,206,252,207,173,254	
:rem 209		:rem 228	
50860 DATA 228,145,251,96,165,251,201		51220 DATA 207,41,3,201,3,208,16	
:rem 41		:rem 20	
50870 DATA 32,240,22,152,24,105,40		51230 DATA 173,253,207,240,11,16,6	
:rem 125		:rem 128	
50880 DATA 168,177,251,201,160,240,1		51240 DATA 238,253,207,76,107,199,206	
:rem 238		:rem 46	
50890 DATA 96,152,56,233,40,168,169		51250 DATA 253,207,173,254,207,41,12	
:rem 213		:rem 233	
50900 DATA 100,145,251,96,152,56,233		51260 DATA 201,12,208,16,173,252,207	
:rem 237		:rem 228	
50910 DATA 120,168,177,251,201,99,240		51270 DATA 240,11,16,6,238,252,207	
:rem 37		:rem 133	
50920 DATA 1,96,152,24,105,120,168		51280 DATA 76,132,199,206,252,207,174	
:rem 136		:rem 46	
50930 DATA 169,100,145,251,96,32,99		51290 DATA 249,207,208,32,174,240,207	
:rem 204		:rem 37	
50940 DATA 119,120,226,249,239,228,160		51300 DATA 224,60,176,25,173,253,207	
:rem 92		:rem 236	
50950 DATA 160,227,247,248,98,121,111		51310 DATA 24,109,1,208,201,80,176	
:rem 42		:rem 128	

51320 DATA 5,169,244,76,191,199,201
:rem 202
51330 DATA 244,144,27,169,80,76,191
:rem 202
51340 DATA 199,173,253,207,24,109,1
:rem 193
51350 DATA 208,201,41,176,5,169,244
:rem 191
51360 DATA 76,191,199,201,244,144,2
:rem 196
51370 DATA 169,41,141,1,208,173,252
:rem 188
51380 DATA 207,48,32,24,109,0,208
:rem 87
51390 DATA 141,240,207,173,249,207,105
:rem 82
51400 DATA 0,141,249,207,201,1,208
:rem 120
51410 DATA 42,173,240,207,201,55,144
:rem 227
51420 DATA 35,32,155,200,76,4,200
:rem 75
51430 DATA 24,109,0,208,141,240,207
:rem 174
51440 DATA 173,249,207,105,255,141,249
:rem 90
51450 DATA 207,208,12,173,240,207,201
:rem 20
51460 DATA 25,176,5,169,25,141,240
:rem 143
51470 DATA 207,173,240,207,141,0,208
:rem 230
51480 DATA 173,16,208,41,254,13,249
:rem 195
51490 DATA 207,141,16,208,173,31,208
:rem 238
51500 DATA 41,1,240,3,76,101,200:rem 13
51510 DATA 96,162,5,189,119,4,201
:rem 95
51520 DATA 57,240,6,254,119,4,76:rem 46
51530 DATA 58,200,169,48,157,119,4
:rem 151
51540 DATA 202,208,235,76,58,200,162
:rem 238
51550 DATA 5,189,39,4,201,48,240:rem 46
51560 DATA 6,222,39,4,76,222,200:rem 36
51570 DATA 169,57,157,39,4,202,208
:rem 155
51580 DATA 235,120,169,234,141,21,3
:rem 183
51590 DATA 169,49,141,20,3,88,169
:rem 110
51600 DATA 255,133,2,76,222,200,0
:rem 72
51610 DATA 162,0,160,240,238,32,208
:rem 177
51620 DATA 232,208,250,200,208,247,169
:rem 81
51630 DATA 0,141,32,208,162,3,189
:rem 84
51640 DATA 39,4,201,48,240,4,222:rem 33
51650 DATA 39,4,96,169,57,157,39:rem 71
51660 DATA 4,202,208,237,162,5,169
:rem 144
51670 DATA 48,157,39,4,202,208,250
:rem 147
51680 DATA 104,104,76,81,200,160,15
:rem 181
51690 DATA 162,3,189,39,4,201,57:rem 52
51700 DATA 240,6,254,39,4,76,180:rem 44



**Shuttle
in fuga**

51710 DATA 200,169,48,157,39,4,202
:rem 143
51720 DATA 208,235,136,208,230,169,0
:rem 238
51730 DATA 141,249,207,169,25,141,240
:rem 35
51740 DATA 207,169,148,141,1,208,172
:rem 244
51750 DATA 156,200,192,9,240,4,136
:rem 139
51760 DATA 140,156,200,173,5,4,201
:rem 127
51770 DATA 160,208,5,169,227,141,5
:rem 146
51780 DATA 4,96,238,224,207,173,224
:rem 203
51790 DATA 207,201,1,240,3,76,124
:rem 82
51800 DATA 201,169,0,141,224,207,173
:rem 229
51810 DATA 27,212,201,7,176,25,168
:rem 141
51820 DATA 185,0,207,201,20,208,8
:rem 80
51830 DATA 169,1,153,16,207,76,16
:rem 97
51840 DATA 201,201,255,208,5,169,255
:rem 241
51850 DATA 153,16,207,160,6,185,0
:rem 89
51860 DATA 207,24,121,16,207,153,0
:rem 130
51870 DATA 207,72,152,10,170,104,157
:rem 236
51880 DATA 1,208,136,208,235,160,6
:rem 142
51890 DATA 185,0,207,201,20,240,10
:rem 124
51900 DATA 201,255,240,6,136,208,242
:rem 232
51910 DATA 76,66,201,169,0,153,16
:rem 94
51920 DATA 207,76,52,201,173,30,208
:rem 187
51930 DATA 41,1,240,51,162,0,160:rem 22
51940 DATA 240,238,32,208,232,208,250
:rem 31
51950 DATA 200,208,247,169,0,141,32
:rem 186
51960 DATA 208,162,3,189,39,4,201
:rem 98
51970 DATA 48,240,6,222,39,4,76 :rem 3
51980 DATA 49,234,169,57,157,39,4
:rem 120
51990 DATA 202,208,235,162,5,169,48
:rem 204
52000 DATA 157,39,4,202,208,250,76
:rem 136
52010 DATA 49,234,256 :rem 13

per tutti coloro che vogliono davvero "dialogare" col computer

FASCICOLI SETTIMANALI
24
IN EDICOLA
DAL 21-9-84

La rapida diffusione dei computer nelle aziende, nelle scuole, nelle case ha reso urgente e indispensabile per molti imparare il linguaggio del calcolatore. Ci sono molti modi per farlo. Il più nuovo, originale, rapido e divertente si chiama...

ABC Personal Computer

l'opera creata per rispondere alle esigenze di chi, per lavoro, studio, hobby vuole acquisire un'effettiva padronanza dei piccoli elaboratori per sfruttarne le immense, entusiasmanti possibilità.

ABC Personal Computer

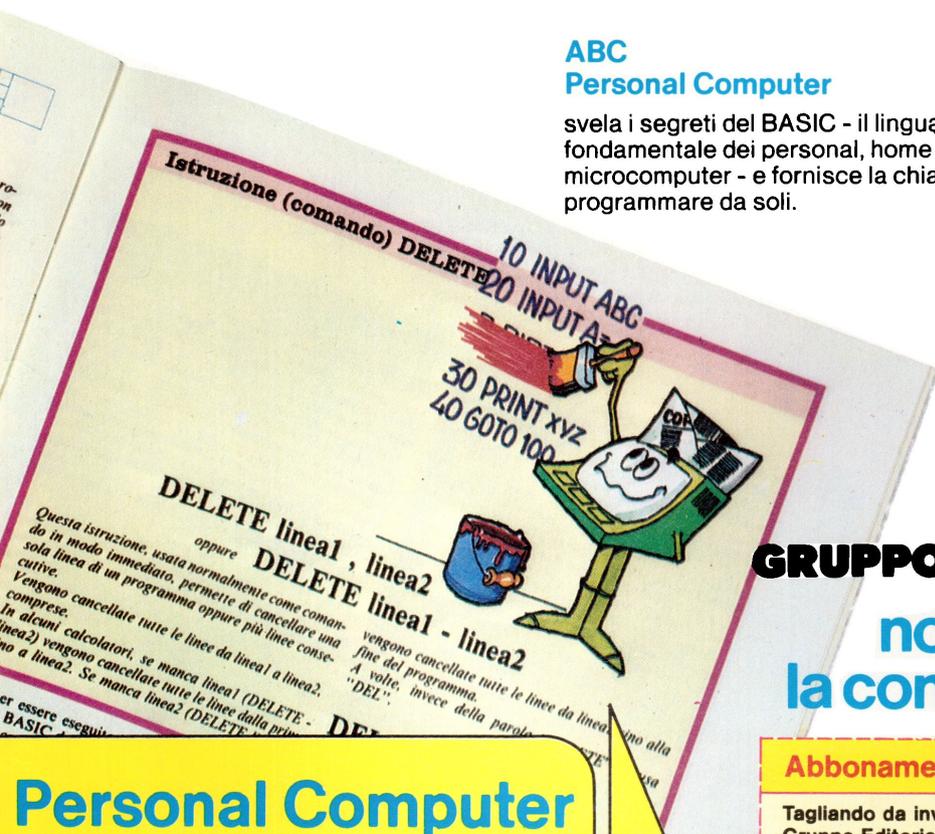
tratta il BASIC in modo diverso, organico, comprensibile a tutti perchè nasce dalla grande esperienza e dalla capacità divulgativa del Gruppo Editoriale Jackson.

ABC Personal Computer

svela i segreti del BASIC - il linguaggio fondamentale dei personal, home e microcomputer - e fornisce la chiave per programmare da soli.

ABC Personal Computer

è un'opera preziosa, **comprensibile per i ragazzi**, perchè comincia proprio dall'abc dell'informatica; **ideale per i professionisti**, perchè rapida, esauriente, concreta; **preziosa per gli appassionati** perchè ricca di programmi già testati, di suggerimenti, di notizie sul mondo dei piccoli computer.



GRUPPO EDITORIALE JACKSON
noi l'informatica
la conosciamo davvero

Abbonamento-risparmio

Tagliando da inviare in busta chiusa a:
Gruppo Editoriale Jackson "ABC Personal Computer"
via Rosellini, 12 - 20124 Milano

Sì, desidero sottoscrivere l'abbonamento risparmio ai 24 fascicoli di ABC Personal Computer e alle copertine dei 4 volumi dell'opera. Tutto al prezzo speciale di L. 80.000 invece di L. 96.000.

Allego alla presente

- assegno non trasferibile di L. 80.000 a voi intestato
- fotocopia di versamento di L. 80.000 sul ccp n. 11666203
- fotocopia di vaglia postale di L. 80.000 a voi intestato

I fascicoli dovranno essere inviati a:

Nome _____ Cognome _____
Via _____
Città _____ Prov. _____ C.A.P. _____
Data _____ Firma _____

Personal Computer

24 appuntamenti in edicola per arricchire la biblioteca di casa con:

- 2 volumi di Lezioni per complessive 608 pagine
- 1 volume di **Computer-test**
- 1 **Dizionario di Informatica** di oltre 208 pagine

L'opera è elegantemente rilegata in similpelle.

Centinaia di illustrazioni a colori, foto e disegni.

Formato dei volumi cm. 21x28.

Formato del Dizionario di Informatica cm. 15x21.





Come aumentare la velocità del VIC

di **D. Carmichael**
trad. ed adatt. di **U. G. Barzagli**

Una delle tecniche di animazione più frequentemente usate nei programmi BASIC consiste nell'utilizzare più istruzioni PRINT per far comparire caratteri sullo schermo. Questo articolo mostra come una routine in linguaggio macchina di semplice utilizzo possa aumentare le velocità del procedimento, e possa addirittura rendere più rapida la risposta della vostra tastiera al di fuori dei programmi.

Se scrivete giochi o programmi che utilizzano la tastiera per muovere oggetti sullo schermo, la mancanza di velocità dovrebbe essere la vostra maggior fonte di problemi. Tuttavia, se utilizzate una istruzione GET per muovere gli oggetti, c'è un modo per aumentare di molto la velocità. Ciò si può ottenere manipolando i "controlli" della tastiera all'interno dello stesso VIC.

I controlli della tastiera - Come funzionano

Innanzitutto, per assicurarvi che il vostro VIC sia in condizioni standard per questa dimostrazione, dovrete spegnerlo e riaccenderlo per azzerarlo. Premete la barra spaziatrice, e continuate a tenerla premuta. Il cursore si muoverà attraverso lo schermo perché il "sistema operativo" — un programma di controllo interno del calcolatore — dà istruzione al VIC di ripetere il procedimento, anche se voi avete premuto la spaziatrice una sola volta. Ora premete un qualsiasi tasto alfabetico, e continuate a tenerlo abbassato. Supponiamo (e, d'ora in avanti, lo faremo anche negli esempi successivi) che voi abbiate abbassato il tasto della "A". Anche se voi continuate a tenerlo premuto, la "A" non verrà ripetuta sullo schermo. Ciò perché gli unici tasti del VIC per i quali la ripetizione sia programmata sono la barra spaziatrice,

INSERT/DELETE, e i due tasti di controllo dei cursori.

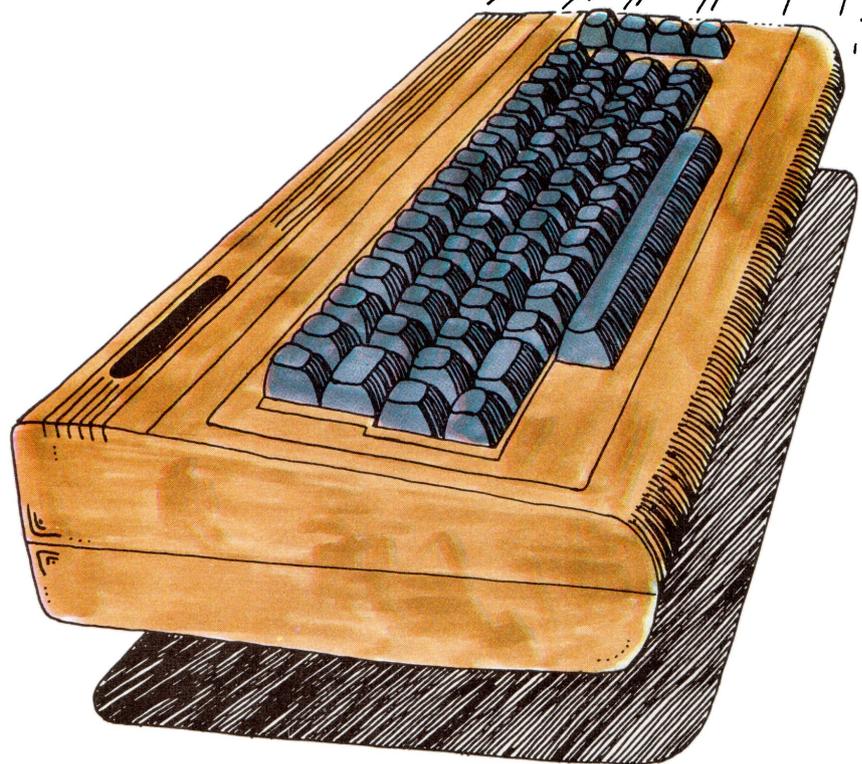
Pulite lo schermo premendo SHIFT-CLR/HOME, quindi battete POKE-650,128. La cella di memoria segnala al sistema operativo quali tasti ripetere. Normalmente è inizializzata a 0, il che permette la sola ripetizione dei tasti suddetti. Un valore di 128 permette la ripetizione di tutti i tasti. In questo caso, se premeste "A", si ripeterebbe come la barra spaziatrice. Lasciate (tramite una istruzione POKE) la cella 650

inizializzata a 128, per ora.

Come aumentare la velocità di ripetizione

Premete e tenete abbassato il tasto della "A" qualche volta, noterete come vi sia una pausa di una frazione di secondo, prima che cominci la ripetizione dei caratteri. La durata di questa pausa è controllata dalla locazione di memoria 652, che non è altro che un "contatore".

Quando voi premete il tasto "A", que-



sta cella contiene un valore numerico di 15. Quindi comincia un conto alla rovescia, con passo unitario, finché non raggiunge lo zero, la qual cosa segnala al calcolatore di iniziare il procedimento di ripetizione.

Per abbreviare la pausa, potreste pensare di inserire con una istruzione di POKE un numero minore di 15 nella locazione di memoria 652. Ma a causa del modo in cui il sistema operativo del VIC lavora ("correndo" attraverso il calcolatore 60 volte al secondo ri-inizializzando diversi parametri ecc.), la locazione 652 verrebbe rapidamente cambiata di nuovo in 15. Tuttavia, potete utilizzare un programma che mantenga questa locazione inizializzata a 0, evitando così la pausa prima della ripetizione. Inserite in memoria il programma 1 — un programma BASIC che ha funzioni di "caricatore", che provvede, cioè, ad inserire tramite istruzioni POKE un breve programma in linguaggio macchina nel buffer del registratore (un'area di memoria utilizzabile). Verificatelo attentamente, quindi mandatelo in esecuzione battendo RUN 59998. Ora il programma in linguaggio macchina è inserito in memoria e pronto per l'uso. Battete SYS828 per farlo partire.

Ora, se premete il tasto della "A", noterete che la ripetizione comincia immediatamente. Con la locazione di memoria 652 costantemente ri-inizializzata a 0, non c'è conto alla rovescia, quindi non c'è nemmeno alcuna pausa.

Per inciso, se commettete errori nell'inserire uno dei programmi in linguaggio macchina e siete costretti a spegnere il VIC ed a riaccenderlo per riprendere, ricordatevi di ripetere l'istruzione POKE 650, 128 prima di proseguire in questo articolo.

Un'altra locazione di memoria che vi aiuterà a velocizzare i vostri programmi è l'indirizzo di memoria 651. Anche

questa cella è un "contatore", fissa il ritardo di tempo tra l'uscita su video di ciascun carattere in caso di ripetizione. La cella 651 usa la stessa tecnica di conto alla rovescia della 652, fatta eccezione per il fatto che comincia da 4 invece che da 15. Durante le ripetizioni, conta alla rovescia fino a 1, stampa il carattere, e quindi conta di nuovo all'indietro. In altre parole, conta "4-3-2-1-PRINT il carattere", "4-3-2-1-PRINT il carattere", ecc.

Come per la locazione 652, potete abbreviare il ritardo di tempo inserendo, con una istruzione POKE un valore inferiore a 4 nella locazione 651. Il sistema operativo prende in esame anche la cella 651, e la ri-inizializza a 4 ogni sessantesimo di secondo. Ma, ancora una volta, possiamo caricare un programma che aggira l'ostacolo.

Battete il programma 2, che è un'altro programma in linguaggio macchina. Questo programma modifica entrambe le locazioni di memoria 651 e 652 e velocizza l'intero procedimento di ripetizione. Verificate attentamente il vostro lavoro di copiatura, quindi battete RUN59997 per inserire il programma in linguaggio macchina in memoria. Battete SYS828 per farlo partire; ora premete la lettera "A" e potrete notare come la ripetizione sia molto più rapida. Paragonatela alla velocità normale premendo RUN/STOP e RESTORE contemporaneamente (per interrompere il programma in linguaggio macchina) e, quindi, premendo nuovamente la "A".

Il programma 2 potrà essere aggiunto ai vostri programmi BASIC e contribuirà ad aumentarne la velocità.

Tuttavia, bisogna far attenzione a due cose circa questo programma. La prima è che, dopo che il programma BASIC iniziale è stato mandato in esecuzione, ed il programma in linguaggio macchina è stato felicemente inserito in memoria dalle istruzioni POKE, po-

tete cancellare le righe 59997-59999. Esse non sono necessarie. La seconda è che, a causa del fatto che il programma in linguaggio macchina utilizza il buffer del registratore, non potete utilizzare il salvataggio su nastro mentre il programma in linguaggio macchina è in esecuzione. Premete RUN/STOP-RESTORE per disabilitarlo prima di usare il registratore.

Il programma 3 è opzionale. È un programma che utilizza la istruzione GET per muovere il carattere "pallina piena" (CHR\$(81)) sullo schermo. È stato scritto per utilizzare la routine in linguaggio macchina che permette di aumentare la velocità. Potete aggiungere anch'esso ai vostri programmi BASIC o costruirli attorno ad esso.

Come usare il programma 3

Battete accuratamente i programmi 2 e 3. Se volete risparmiare memoria cambiando i numeri di riga del programma 3, potete farlo, ma ricordatevi di cambiare i GOTO corrispondenti, ecc. Se volete cambiare il carattere che viene spostato lungo lo schermo, esso è controllato dalla variabile H in riga 6000 (fate riferimento al "Vic user's manual", per sapere il numero di POKE corrispondente al carattere voluto). Per eseguire il programma, prima battete RUN59997 per inserire il programma in linguaggio macchina di supporto in memoria, quindi battete RUN 60000 per eseguire il programma BASIC. I tasti seguenti controllano gli spostamenti del carattere:

- W = verso sinistra ←
- R = verso destra →
- P = verso l'alto ↑
- . = verso il basso ↓

La riga 60050 è opzionale. Senza di essa, il movimento della pallina sarà leggermente più veloce, ma la pallina lampeggerà, la qual cosa potreste trovare più o meno gradevole.

Programma 1

```
59998 FORA=828TO848:READB:POKEA,B:NEXT:EN
D                                     :rem 160
59999 DATA120,169,73,141,20,3,169,3,141,2
1,3,88,96,169,0,141,140,2,76,191,23
4                                     :rem 23
```

Programma 2

```
59997 FORA=828TO853:READB:POKEA,B:NEXT:EN
D                                     :rem 155
59998 DATA120,169,73,141,20,3,169,3,141,2
1,3,88,96,169,1,141,139,2,169,0,141
,140,2,76                             :rem 156
59999 DATA191,234                     :rem 147
```

Programma 3

```
59997 FORA=828TO853:READB:POKEA,B:NEXT:EN
D                                     :rem 155
```

```
59998 DATA120,169,73,141,20,3,169,3,141,2
1,3,88,96,169,1,141,139,2,169,0,141
,140,2,76                             :rem 156
59999 DATA191,234                     :rem 147
60000 A=7680:B=38400:C=8185:E=A:F=38905:G
=B:H=81:I=32:POKE650,128:SYS828:PRI
NT"{CLR}":D=1                           :rem 209
60010 GETA$:IFA$="R"THEND=1:GOTO60060
                                           :rem 141
60020 IFA$="W"THEND=-1:GOTO60060:rem 65
60030 IFA$="*"THEND=22:GOTO60060:rem 27
60040 IFA$="P"THEND=-22:GOTO60060
                                           :rem 111
60050 IFD=0THEN60010                    :rem 97
60060 POKEA,I:A=A+D:B=B+D:POKEA,H:POKEB,6
:D=0                                     :rem 201
60070 IFA<ETHENA=C:B=F:GOTO60060:rem 172
60080 IFA>CTHENA=E:B=G:GOTO60060:rem 176
60090 GOTO60010                          :rem 47
```



Dott. Video 64

di D. W. Martin
trad. ed adatt. di U. Barzaghi

I tasti di controllo cursore del vostro C64 vi danno già alcune delle più potenti funzioni di schermo disponibili per un home computer, ma questo programma di utilità pratica ne aggiunge altre: cancella lo schermo al di sotto del cursore, lo cancella al di sopra, e posiziona il cursore all'angolo inferiore sinistro dello schermo, il tutto con la velocità del linguaggio macchina.

Nel corso della revisione di lunghi programmi o facendo ripetuti calcoli numerici in modo immediato (cioè al di fuori da un programma BASIC) è spesso utile poter cancellare una parte dello schermo lasciando la rimanente intatta. È anche utile, a volte, poter posizionare il cursore nell'angolo inferiore sinistro dello schermo invece che nella normale posizione superiore sinistra ottenibile premendo il tasto HOME.

Benché la Commodore abbia inserito eccellenti funzioni di schermo nel C64, "Dott. Video 64" aggiunge una maggior flessibilità fornendovi altri tre tasti di controllo cursore. Una tecnica particolare permette a Dott. Video di funzionare anche quando state battendo o mentre è in esecuzione un altro programma. Dal momento che il programma è interamente scritto in linguaggio macchina, non occupa nessuna parte della memoria normalmente usata per la programmazione BASIC. Le nuove funzioni di controllo del cursore sono assegnate a tre dei tasti di funzione del C64. Gli assegnamenti sono i seguenti:

F1 Cancella lo schermo fino al margine superiore, cominciando con la riga contenente il cursore.

F3 Cancella lo schermo fino al margine inferiore, a partire dalla linea contenente il cursore.

F5 Pone il cursore nell'angolo inferiore sinistro dello schermo.

Come opera il "Dott. Video 64"

Ogni sessantesimo di secondo il vostro C64 si arresta, qualsiasi cosa stia facendo, e passa un po' di tempo a leggere la tastiera ed a sbrigare altri compiti di economia domestica. Queste so-

ste vengono chiamate "interruzioni" o "interrupt", ed il programma in linguaggio macchina in esecuzione durante le interruzioni (segnale IRQ), va a guardare in una coppia di locazioni di memoria per trovare l'indirizzo di partenza (chiamato vettore IRQ) del sottoprogramma di servizio delle interruzioni che deve essere eseguito. Nel C64, il vettore IRQ è contenuto nelle locazioni 788 e 789, che identificano normalmente l'indirizzo 59953, l'inizio del sottoprogramma IRQ nella ROM (memoria a sola lettura e perciò inalterabile) standard. Tuttavia, poiché il vettore IRQ è memorizzato nella memoria ad accesso casuale (RAM), che è modificabile, possiamo sostituire l'indirizzo del nostro personale sottoprogramma in linguaggio macchina ed aggiungerlo alla normale routine di servizio delle interruzioni.

Come tutte le routine attivate dalle interruzioni, Dott. Video 64 viene eseguito finché non ri-inizializzate il vo-

stro calcolatore (premendo RUN/STOP e RESTORE contemporaneamente, ad esempio). Non viene disattivato premendo il solo tasto STOP. Dopo aver ri-inizializzato, potete riattivare i nuovi tasti di gestione dello schermo battendo SYS49152.

Come inserire il programma

Dott. Video è un programma in linguaggio macchina che utilizza un caricatore BASIC per inserire tramite istruzioni POKE i dati in memoria, e lancia la istruzione SYS che attiva il programma. Una somma di controllo viene calcolata in modo da aiutarvi ad identificare errori di battitura eventuali nelle istruzioni DATA ma, dal momento che il programma caricatore si cancella da sé (con una istruzione NEW) in modo da non intralciare l'area di memoria del BASIC, dovete badare a registrarne una copia prima di mandarlo in esecuzione con una istruzione RUN per la prima volta.

```
Dott.Video 64 200 FORI=0TO148 :rem 111
                210 READJ:POKE49152+I,J:X=X+J:NEXTI :rem 175
                230 IFX<>17524THENPRINT"ERRORE NELLE ISTR
                    UZIONI DATA":STOP :rem 242
                240 SYS49152:NEW :rem 190
                300 DATA120,169,13,141,20,3,169,192,141,2
                    1,3,88,96,165,197,41 :rem 236
                310 DATA127,201,4,208,27,169,0,133,25,169
                    ,4,133,26,216,24,165 :rem 226
                320 DATA209,105,40,133,27,165,210,133,28,
                    144,2,230,28,24,144,46 :rem 59
                330 DATA165,197,41,127,201,5,208,19,165,2
                    09,133,25,165,210,133,26 :rem 175
                340 DATA169,231,133,27,169,7,133,28,24,14
                    4,19,201,6,208,67,169 :rem 44
                350 DATA192,133,209,169,7,133,210,169,24,
                    133,214,24,144,44,216,56 :rem 180
                360 DATA165,27,229,25,133,29,165,28,229,2
                    6,133,30,169,32,166,30 :rem 92
                370 DATA240,12,160,0,145,25,200,208,251,2
                    30,26,202,208,246,166,29 :rem 156
                380 DATA240,8,160,0,145,25,200,202,208,25
                    0,169,0,133,211,169,32 :rem 54
                390 DATA133,197,76,49,234 :rem 17
```

VIC sound

C64 Sound Tester

di R.V. Picard

trad. ed adatt. di M. Cristuibu Grizzi

Le capacità sonore del C64 superano ampiamente le possibilità di tutti i precedenti microcomputer. Prima dell'avvento del C64, la possibilità di scegliere forma d'onda, attack/decay e sustain/release di un suono esistevano solo per i sintetizzatori sonori. D'altra parte, capire e regolare tutti i vari parametri di un suono può creare perplessità sia programmatori principianti che a quelli più esperti.

Questo programma permette di fare esperimenti con tutti questi parametri, ascoltare il risultato ed eventualmente,

modificare qualche regolazione del suono.

Si possono esplorare tutte le otto ottave ed il generatore di rumore, ed in qualsiasi momento si possono controllare i parametri di regolazione usati prima di settarne degli altri.

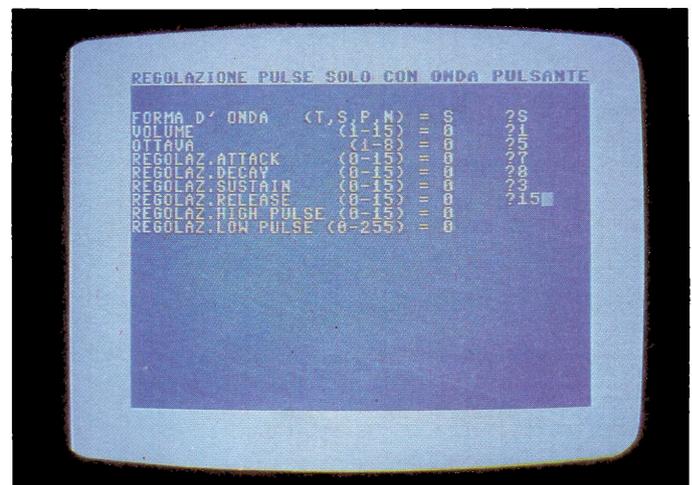
Eseguendo il programma, apparirà un elenco dei valori correnti di regolazione, con un cursore in prossimità del primo. Se si desidera modificare il valore di quella linea, basta digitare il nuovo valore seguito dal RETURN; mentre se non lo si desidera modifica-

re basta premere soltanto il RETURN. Le lettere 'T', 'S' e 'P' si riferiscono alle forme d'onda rispettivamente triangolare, a dente di sega e pulsante. 'N' si riferisce al generatore di rumore, e 'Q' provoca l'uscita dal programma. Ogni volta che si desidera controllare i parametri basta battere 'D'.

Dopo che sia stato immesso l'ultimo valore, verrà generato un suono parametrizzato nei valori dati, dopo il quale il programma riprenderà da capo.

C64 Sound Tester

```
5 DIMSO(16,8) :rem 205
10 HF=54273:LF=54272:AD=54277:SR=54278:W=
  54276:V=54296:HP=54275:LP=54274
   :rem 69
15 FORI=1TO8:A(I)=0:NEXT:W$="S" :rem 99
20 FORO=1TO8:FORN=1TO16:READSO(N,O):NEXT:
  NEXT :rem 204
25 FORN=1TO8:READD(N):NEXT :rem 151
100 PRINT"{CLR}";CHR$(18);"REGOLAZIONE PU
  LSE SOLO CON ONDA PULSANTE";CHR$(146)
   :rem 5
102 PRINT :rem 32
110 PRINT"FORMA D' ONDA{ 3 SPAZI}(T,S,P,N
  ) = ";W$: :rem 114
112 PRINT"VOLUME{ 13 SPAZI}(1-15) =";A(1)
   :rem 141
114 PRINT"OTTAVA{ 14 SPAZI}(1-8) =";A(2)
   :rem 89
116 PRINT"REGOLAZ.ATTACK{ 5 SPAZI}(0-15)
  =";A(3) :rem 180
118 PRINT"REGOLAZ.DECAY{ 6 SPAZI}(0-15) =
  ";A(4) :rem 101
120 PRINT"REGOLAZ.SUSTAIN{ 4 SPAZI}(0-15)
  =";A(5) :rem 32
122 PRINT"REGOLAZ.RELEASE{ 4 SPAZI}(0-15)
  =";A(6) :rem 253
124 PRINT"REGOLAZ.HIGH PULSE (0-15) =";A(
  7) :rem 168
126 PRINT"REGOLAZ.LOW PULSE (0-255) =";A(
  8) :rem 179
130 PRINT"{HOME}";"{ 3 GIU'"; :rem 101
140 PRINTTAB(33);"?";:GOSUB500:IFZ$="D"TH
  EN600 :rem 201
```



```
150 IFZ$="Q"THENPRINT"{CLR}":END:rem 25
155 IFZ$<>CHR$(13)THENW$=Z$ :rem 61
160 FORO=1TO6 :rem 20
170 PRINTTAB(33);"?";:GOSUB500:IFZ$<>CHR$
  (13)THENA(O)=VAL(Z$) :rem 113
180 NEXT :rem 216
190 IFW$<>"P"THEN230 :rem 118
200 FORO=7TO8 :rem 23
210 PRINTTAB(33);"?";:GOSUB500:IFZ$<>CHR$
  (13)THENA(O)=VAL(Z$) :rem 108
220 NEXT :rem 211
230 POKEV,A(1) :rem 9
```

```

240 POKEAD,16*A(3)+A(4):POKESR,16*A(5)+A(6) :rem 66
250 IFW$="T"THENPOKEW,17 :rem 191
260 IFW$="S"THENPOKEW,33 :rem 189
270 IFW$="P"THENPOKEW,65:POKEHP,A(7):POKE
LP,A(8) :rem 177
280 IFW$="N"THENPOKEW,129 :rem 240
300 O=0 :rem 79
310 FORI=1TO15STEP2:POKEHF,SO(I,A(2)):POK
ELF,SO(I+1,A(2)):O=O+1:FORN=1TOD(O)
:rem 124
311 NEXT :rem 212
315 POKEHF,0:POKELF,0:NEXT:POKEW,0:POKEAD
,0:POKESR,0:POKEHP,0:POKELP,0
:rem 16
320 GOTO100 :rem 95
500 Z$="" :rem 148
510 GETY$:PRINTCHR$(18);" ";CHR$(146);:FO
RI=1TO25:NEXT:PRINTCHR$(157);" ";
:rem 34
515 PRINTCHR$(157);:FORI=1TO25:NEXT:IFY$=
""THEN510 :rem 57
520 PRINTY$; :rem 220
530 IFY$=CHR$(13) THENIFLEN(Z$)=0THENZ$=CH
R$(13) :rem 16
540 IFY$=CHR$(13) THENRETURN :rem 167
550 Z$=Z$+Y$:GOTO510 :rem 132
600 PRINT"{CLR}" :rem 250
610 PRINT:PRINT"ATTACK/DECAY{ 4 SPAZI}=";
16*A(3)+A(4) :rem 59
620 PRINT"SUSTAIN/RELEASE =";16*A(5)+A(6)
:rem 131
630 PRINT:PRINT:PRINT"PREMI UN TASTO PER
CONTINUARE"; :rem 189
640 GETY$:IFY$=""THEN640 :rem 135
650 GOTO100 :rem 101
1000 DATA1,155,1,90,1,110,1,155,1,90,1,11
0,1,155,1,205 :rem 85
1010 DATA3,54,2,179,2,220,3,54,2,179,2,22
0,3,54,3,155 :rem 68
1020 DATA6,108,5,103,5,185,6,108,5,103,5,
185,6,108,7,53 :rem 181
1030 DATA12,216,10,205,11,114,12,216,10,2
05,11,114,12,216,14,107 :rem 67
1040 DATA25,177,21,154,22,227,25,177,21,1
54,22,227,25,177,28,214 :rem 126
1050 DATA51,97,43,52,45,198,51,97,43,52,4
5,198,51,97,57,172 :rem 174
1060 DATA102,194,86,105,91,140,102,194,86
,105,91,140,102,194,115,88 :rem 21
1070 DATA205,133,172,210,183,25,205,133,1
72,210,183,25,205,133,230,176
:rem 147
1080 DATA500,250,250,250,250,250,250,1000
:rem 247

```

Yesterday

di P. Ferrari

Ecco un altro arrangiamento di un "classico" per il Commodore 64 ed il Music Editor pubblicato sul primo numero della rivista. Questa volta si tratta però di un "classico" degli anni '70: la bellissima "Yester-

day' dei Beatles. Ricordate tuttavia che lo scopo principale di questa rubrica è quello di stimolare la vostra fantasia e creatività a sperimentare le capacità musicale del Commodore 64.

Utilizzate quindi l'Editor non solo per "suonare" gli spartiti pubblicati, ma anche per scrivere i vostri spartiti e per creare delle sonorità originali.

YESTERDAY

					007	08	C 5	A 4		027	08	F 6	F 5	
DC1:	240	240	244	008	08	C 5	A 4			028	08	E 6	E 5	D 3
DC2:	008	008	012	009	08	C 5	A 4	F 3		029	04	D 6	D 5	
W:	065	065	065	010	08	C 5	A 4			030	32	D 6	D 5	
A/D:	064	064	064	011	08	C 5	A 4			031	16			C 3
S/R:	127	253	255	012	08	C 5	A 4			032	32	***	F 5	A#2
C1:	0			013	08	C 5	A 4			033	08	D 6		
C2:	000			014	08	C 5	A 4			034	08	D 6		
FOH:	000			015	08	C 5	A 4			035	08	C 6	E 5	C 3
LEV:	015			016	08	C 5	A 4			036	08	A#5		
				017	08	G 5	A 4	F 2		037	08	A 5		
				018	08	F 5				038	08	G 5		
				019	32	F 5				039	16	A#5	C 5	F 3
				020	08			F 3		040	08	A 5		
001	08	C 5	A 4	F 3	021	32	***	B 4	E 3	041	08	A 5		
002	08	C 5	A 4		022	08	A 5	A 4		042	16	A 5	C 5	F 3
003	08	C 5	A 4		023	08	B 5	B 4		043	16	G 5	C 5	E 3
004	08	C 5	A 4		024	08	C#6	C#5	A 2	044	16	F 5	D 5	D 3
005	08	C 5	A 4		025	08	D 6	D 5		045	08	A 5	C 5	
006	08	C 5	A 4		026	08	E 6	E 5		046	08	G 5	B 4	

047	24	G 5	B 4	G 2	087	16	F 5	A#4	A#2	127	08	C#6	C#5	A 2
048	08	D 5	D 5		088	08	A 5	C 5	F 2	128	08	D 6	D 5	
049	16	F 5	A#4	A#2	089	08	A 5	C 5	F 2	129	08	E 6	E 5	
050	08	A 5	C 5	F 2	090	16	A 5	C 5	F 2	130	08	F 6	F 5	
051	08	A 5	C 5	F 2	091	16			C 3	131	12	E 6	E 5	D 3
052	16	A 5	C 5	F 2	092	32	A 5	A 4	E 3	132	04	D 6	D 5	
053	16			C 3	093	32	A 5	A 4	C#3	133	32	D 6	D 5	
054	08	G 5	C 5	F 2	094	16	D 6	D 5	D 3	134	16			C 3
055	08	F 5			095	16	E 6	E 5	C 3	135	32	***	D 5	A#2
056	32	F 5			096	16	F 6	F 5	A#2	136	08	D 6		
057	16			F 3	097	08	E 6	E 5	A 2	137	08	D 6		
058	32	***	B 4	E 3	098	08	D 6	D 5		138	08	C 6	E 5	C 3
059	08	A 5	A 4		099	24	E 6	E 5	G 2	139	08	A#5		
060	08	B 5	B 4		100	08	D 6			140	08	A 5		
061	08	C#6	C#5	A 2	101	16	C 6	E 5	C 3	141	08	G 5		
062	08	D 6	D 5		102	16	D 6			142	16	A#5	F 5	F 3
063	08	E 6	E 5		103	32	A 5	F 5	F 2	143	08	A 5		
064	08	F 6	F 5		104	32			F 3	144	08	A 5		
065	12	E 6	E 5	D 3	105	32	A 5	A 4	B 2	145	16	A 5		F 3
066	04	D 6	D 5		106	32	A 5	A 4	C#3	146	16	G 5	C 5	E 3
067	16	D 6	D 5		107	16	D 6	D 5	D 3	147	16	F 5	D 5	D 3
068	16			D 3	108	16	E 6	E 5	C 3	148	08	A 5	C 5	
069	16			C 3	109	16	F 6	F 5	A#2	149	08	G 5	B 4	
070	32	***	F 5	A#2	110	08	E 6	E 5	A 2	150	24	G 5	B 4	G 2
071	08	D 6			111	08	D 6			151	08	D 5	D 5	
072	08	D 6			112	24	E 6	E 5	G 2	152	16	F 5	D 5	A#2
073	08	C 6	E 5	C 3	113	08	D 6			153	08	A 5	C 5	F 2
074	08	A#5			114	16	C 6	E 5	C 3	154	08	A 5	C 5	F 2
075	08	A 5			115	16	E 6			155	16	A 5	C 5	F 2
076	08	G 5			116	16	F 6	F 5	F 3	156	16			C 3
077	16	A#5	C 5	F 3	117	16		C 5		157	20	F 5	F 5	C 3
078	08	A 5			118	16		A#4		158	20	A 5	A 5	
079	08	A 5			119	16		A 4		159	24	G 5	G 5	B 2
080	16	A 5		F 3	120	08	G 5	G 4	F 2	160	24	D 5	D 5	
081	16	G 5	C 5	E 3	121	08	F 5	F 4		161	28	F 5	D 5	A#2
082	16	F 5	D 5	D 3	122	32	F 5	F 4		162	08	A 5	C 5	F 2
083	08	A 5	C 5		123	16			F 3	163	08	A 5	C 5	F 2
084	08	G 5	B 4		124	32	***	B 4	E 3	164	44	A 5	C 5	F 2
085	24	G 5	B 4	G 2	125	08	A 5	A 4		165	64	A 5	C 5	F 2
086	08	D 5			126	08	B 5	B 4						

L'unico mass-media
di strumenti musicali
e audio-registrazione.

strumenti
MUSICALI



**GRUPPO
EDITORIALE
JACKSON**

QUANDO JACKSON NON È SOLO ELETTRONICA E INFORMATICA.

Recensioni Software

di M. Cristulb Grizzi

Borsa per C64

Dopo il grande successo riscontrato per "Melopoli", la simulazione del famoso Monopoli su Apple II, iniziano ad apparire i primi giochi "di società" computerizzati anche per il C64. "Borsa" è un simpaticissimo programma che permette di passare piacevoli serate in compagnia, stimolando l'intelligenza e... il senso degli affari.

Possono partecipare al gioco da due ad otto giocatori, a ciascuno dei quali viene assegnata inizialmente la dotazione di una carta di credito, un capitale di 100.000 dollari, ed un totale di cinque azioni estratte a sorte. Il percorso del gioco è formato da quaranta diverse caselle, ognuna rappresentante differenti situazioni del mercato azionario, mentre la tecnica del gioco riflette abbastanza realmente le operazioni tipiche del mondo della borsa.

Acquisto e vendita di grossi pacchetti azionari, rialzo o ribasso improvviso delle quotazioni, la scalata ai pacchetti azionari di maggioranza, informazioni e previsioni circa i successi o insuccessi delle varie industrie, il fallimento delle aziende con le sue conseguenze, sono tutte cose con le quali si deve fare i conti per riuscire ad incrementare il proprio capitale fino ad un milione di dollari, quindici azioni ed una carta di credito. Il primo giocatore che rag-



giunge questa meta è dichiarato vincitore, ed è sicuro il suo fiuto negli affari. Il gioco si sviluppa in molteplici schermate e prospetti, nei quali vengono mostrate la propria situazione finanziaria, il listino di borsa, casella in cui ci si trova, etc. Per creare anche una componente di casualità nel gioco, è previsto il lancio dei dadi (dadi elettronici, naturalmente) e 20 tipi diversi di telex che ogni tanto vengono ricevuti dai singoli giocatori, recando buone o cattive notizie da parte degli agenti di borsa che sono stati sguinzagliati.

Il programma è veramente completo e, a differenza di altri giochi per computer, non è dominato dalla casualità: il giocatore è sempre dotato della possibilità di scelta tra lancio dei dadi ed



utilizzo di una tessera della quale è precedentemente entrato in possesso. Dopo anni di totale dominio di alieni e marzianini nel settore dei giochi per computer, stiamo finalmente assistendo alla nascita di giochi un po' meno istintivi ed un po' più intelligenti: speriamo si continui su questa strada.

Borsa per C64 cod. CCAMC01
è in vendita presso:
TECHNOCLUB
Via Rosellini, 12
20124 Milano
a L. 48.000.

LIGHT PEN FLEXIDRAW PER COMMODORE 64

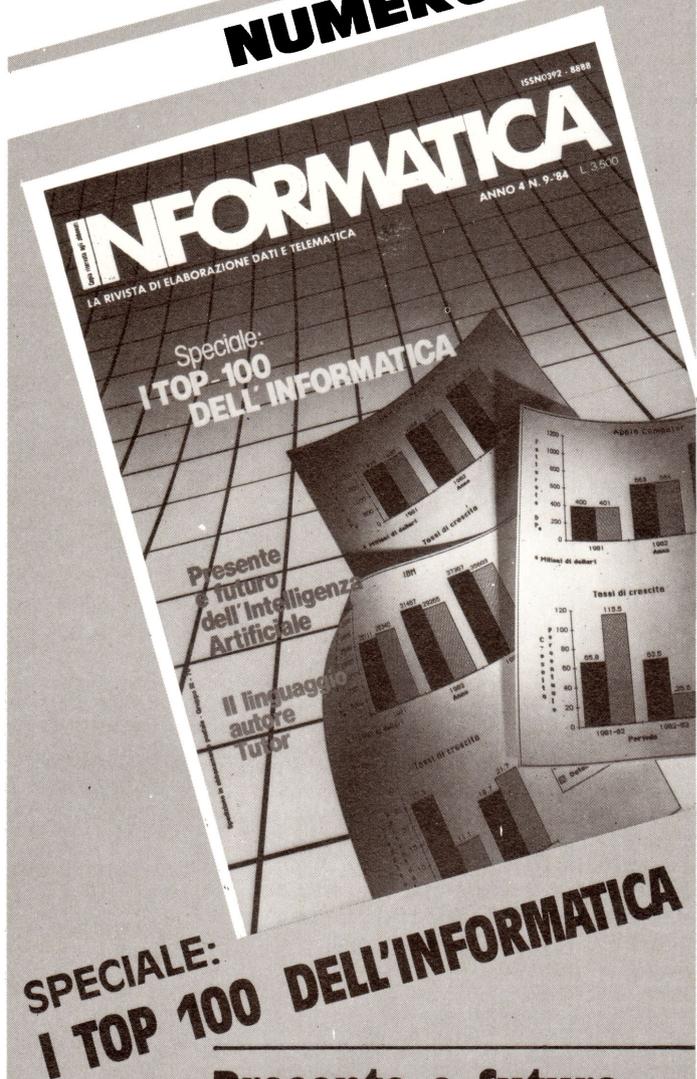
- La GBC Italiana distribuisce una nuova penna ottica per il Commodore 64.
- La penna Flexidraw è un sistema differente e con concetti nuovi rispetto alle light-pen già presenti sul mercato.
- Flexidraw system è uno strumento sviluppato per applicazioni professionali.
- Può essere usato da programmatori ,disegnatori , artisti e per tutte le altre necessità dove è richiesta una periferica d'ingresso con eccellente qualità in alta risoluzione grafica.
- Flexidraw è un package completo con un ricco corredo di software versatile.
- Sul dischetto master di base sono contenuti numerosi programmi : la stesura di schemi elettrici , disegni di architettura , una espressione matematica , l'alfabeto greco , tutti i simboli musicali , un programma dimostrativo del sintetizzatore musicale ed un programma che simula la tastiera di un pianoforte.
- Sono contenuti inoltre quattro spettacolari esempi di grafica.



Cod. SM/3100-12

DISTRIBUITA DALLA GBC

**E' IN EDICOLA
IL NUOVO
NUMERO DI**



**SPECIALE:
I TOP 100 DELL'INFORMATICA**

**Presente e futuro
dell'Intelligenza
artificiale**

Il Linguaggio Tutor



Una pubblicazione
firmata...

GRUPPO EDITORIALE JACKSON s.r.l

20124 Milano - Via Rosellini, 12 - Tel. 02/68.80.951-2-3-4-5
Telex: 333436 GEJ IT

P R I N T

Recensioni Libri

a cura della **Redazione**

Software 64 **practical programs for** **the Commodore 64** (Programmi pratici per il Commodore 64) di **Owen Bishop**

Tredici programmi per il C64 fondamentalmente ispirati ad un uso del computer come organizzatore "casalingo" dei propri dati, scelte ed affari. Non mancano però anche programmi di "relax", quali "Sounds Incredibile" che fornisce una delle migliori dimostrazioni delle potenzialità musicali sessantaquattresche, o "Gottit!" che trasforma il computer in un inflessibile esaminatore della vostra cultura generale. Un altro programma particolarmente utile per il vostro portafogli è "Phone Call Coster", che permette di

calcolare in tempo reale il costo delle vostre telefonate (vedrete che dopo averlo eseguito vi tratterete meno al telefono) e controllare le bollette che vi arrivano. Gli altri programmi, come già accennato, sono indirizzati ad un uso "intelligente" del computer: database per la gestione completa di informazioni, controllo degli stock per la gestione di un magazzino, analisi di mercato per previsioni sulle vendite o sull'andamento di un particolare prodotto, etc.

Questo libro è in versione originale inglese, comunque molto chiaro nel testo ed un po' meno nei listati che sono stati ricavati tramite la stampante Commodore VIC 1515, notoriamente non tra le migliori in fatto di leggibilità.

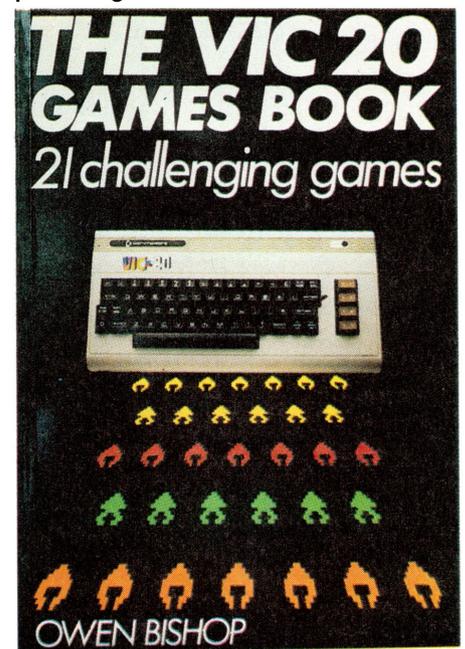
The VIC 20 games book (Il libro dei giochi per il VIC 20)

di **Owen Bishop**

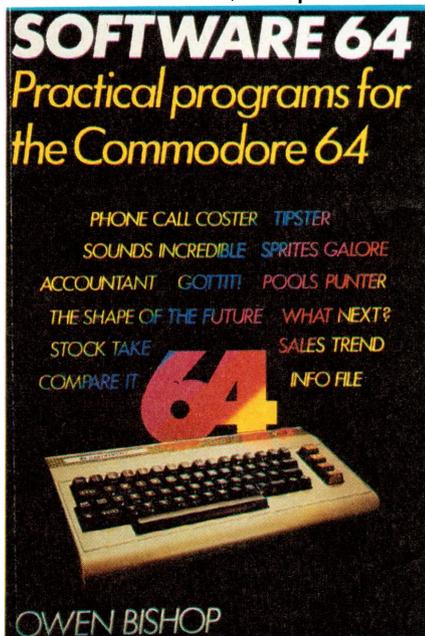
Questa volta ventuno programmi per il VIC 20, tutti orientati verso il divertimento. Girano tutti quanti sul VIC in configurazione base, e non sono quindi richieste espansioni di memoria. Si passa dai giochi nei quali è richiesta prontezza di riflessi e dita velocissime, ad altri più "meditativi" in cui si misura la propria logica con quella del computer.

Un ottimo libro per il principiante che può facilmente imparare come viene strutturato un computer game, e può eventualmente trarre spunti dai programmi proposti per crearne di propri o per adattarli e modificarli secondo le proprie esigenze. Tutti i programmi proposti sono ricchi di colori, di animazioni e di effetti sonori; all'inizio del libro c'è inoltre una parte completa-

mente dedicata alle istruzioni per la perfetta copiatura dei listati, con una chiara spiegazione dei simboli grafici del VIC e delle principali istruzioni BASIC. Anche per questo libro valgono le considerazioni fatte per il precedente circa la notevole chiarezza del testo contrapposta almeno in parte alla non perfetta grafia dei listati.



I libri possono essere ordinati utilizzando il coupon a pag. 82



E' IN EDICOLA IL NUMERO DUE!

SERVIZIO
SPECIALE:

PC-JUNIOR

- **PC SOFTEST:**
Lotus 1-2-3
- **PC-ABC:**
Sistemi operativi
- **PC HARDWARE:**
Il Personal computer XT
- **PC LISTING:**
Calcola l'equo canone
- **GRAFICA CON IL PC**
- **COME E' NATO
L'OLIVETTI M24**
- **PC BOYER'S GUIDE:**
Tutti i sistemi MS-DOS

PC MAGAZINE

Una pubblicazione firmata...



**GRUPPO
EDITORIALE
JACKSON**

ABBONATEVI ADESSO!

10 numeri della rivista Vi costeranno 40.000 lire (anziché 50.000).

Per sottoscrivere l'abbonamento basta compilare il modulo pubblicato qui sotto e spedirlo a:
Gruppo Editoriale Jackson S.r.l.

Via Rosellini, 12 - 20124 Milano,
allegando assegno o fotocopia della ricevuta di versamento sul c/cp n. 11666203 intestato a
GRUPPO EDITORIALE JACKSON - MILANO.

Sì, desidero sottoscrivere un abbonamento PC Magazine (10 numeri al prezzo di Lire 40.000, anziché 50.000)

Allego assegno

Ho effettuato versamento su c/cp n. 11666203

Nome

Cognome

Azienda

Via

CAP Città

Sono in possesso di un PC (o compatibile)

Modello

Non possiedo PC, ma lo utilizzo sul luogo di lavoro

input.output

Piccoli
annunci



Compro, vendo, cambio cartucce per Atari 2600, cartucce per Colecovision e cassette con giochi per VIC 20 in linguaggio macchina.
Riccardo Pedrini - via G. Chiassi, 12/C - 25100 Brescia - Tel.030/304387

Vendo cassetta C64 contenente cinque giochi: Slalom, Briscola, Domino, Macchina Mangiasoldi; e un programma di utility cioè bilancio familiare. Tutto a L.10.000. Scrivere per ricevere la cassetta.
Salvatore Purgato - C.so Europa, 26 - 81030 S. Marcellino (CS) - Tel.081/8121276

Vendo console e quattro cassette Atari (gioco).
Herbert Neumann - via Sulmona, 23 - 20139 Milano - Tel.562895 ore serali.

Compro espansione di memoria 16K per VIC 20
Mario Boccia - V.le America Latina, 113 - 03100 Frosinone - Tel.857270

Cassetta con Puck-Man, Galaxia, Blitz, Pitone, Asteroids, Invaders, S. Invaders, Attacco Alieno, Dollaro, Briscola, Sette e mezzo, Tot.13, Bonzo, Sub Chase, ecc. a L.15.000 un prezzo da sballo!!! Sono 25 giochi!
Enea Meroni - via Como, 5 - 22036 Erba - Tel. 031/611395

Vendo per CBM 64 programmi di Tennis, Wargate, Totocalcio, Attack of Camel ecc. Vendo anche libri per Commodore VIC 20 e cose varie e cassette con vari programmi.
Renato Petroniro - via Lucera, 119 - 71100 Foggia - Tel.0881/43373

Compro, cambio, vendo programmi per CBM 64 di vario genere. Dispongo di splendidi giochi e ottimi programmi di utilità. Spedite le vostre liste per scambi. Rispondo a tutti. Annuncio sempre valido!!!
Roberto Cordovani - C.P.7174 - 47700 Forlì

Cartuccia "Jupiter Lander" per C64. Vendo a L.38.000 o cambio con altra cartuccia che sia pari ottime condizioni.
Giovanni Addabo - via Monte Vettore, 46 - 60131 Ancona

Cambio programmi per VIC 20 inespanso.
Riccardo D'Auria - via Posillipo, 203 - 80123 Napoli

Per VIC 20, vendo una cassetta contenente 10 splendidi giochi (Blitz, Raid On, Isram, ecc.) a sole L.6.000. Affrettatevi è un'occasione da non perdere.
Federico Formia - S.da S.Vincenzo,40/9 - 10131 Torino - Tel.011/6503285

Vendo Cartridge Adventureland per VIC 20 causa doppio regalo. Mai usata.
Alessio Cavalli - via D. Morchio, 1/35/A - 16162 Bolzaneto (GE) - Tel.406054

Cambio, vendo circa 200 programmi per il CBM 64, dispongo inoltre di circa 1.000 programmi per il VIC 20. Per contatti scrivere o telefonare a:
Lionello Zanella - via Virgilio, 21 - 74025 Marina di Ginosa (TA) - Tel. 099/627090

Cambio software per C64 (giochi, utilities). Inviare lista, si risponde a tutti.
Alberto M. Campus - via Genova, 40 - 07100 Sassari - 079/270126

Cambio software per Commodore 64. Vendo anche giochi L. 2.000 (un affare!)
Francesco Billari - via Ciovasso, 4 - 20121 Milano - 870951

Vendo per VIC 20 cartuccia Cosmic Cruncher (VIC-1922), cartuccia Sargon II Chess (VIC-1919), cartuccia Alien (VIC 1906), Cassetta Alien Blitz (VIC-6619)
Carlo Mosconi - Str. Valsalice 227/7/G - 10131 Torino - Tel.8610545

Vendo, cambio, software per C64 (solo Milano)
Telefonare a:
Vincenzo Arcidiacono - P.zza Gramsci, 3 - 20154 Milano - Tel.317676 ore pasti

Cambio, vendo, compro, software per Commodore 64. Richiedere e inviare liste a:
Fabrizio Macellini - via Zanzi,4 - 21037 Lavena P.Teresa (VA) - Tel.0332/550746

Vendo eccezionali programmi su disco per Commodore 64 (650) tutte le novità (Magic Desk, Koala, Superbase, ecc.) a prezzi eccezionali, scrivere, scrivere, scrivere...
Augusto Proietti - via Casilina, 522 - 00100 Roma - Tel. 06/270865

Cambio solo zona Napoli programmi utility e giochi per C64. Vendo base Intellivision + Computer Keyboard
Gianluca Casaburi - via Posillipo, 146 - 80123 Napoli - Tel.081/7694597

Vendo software VIC 20 giochi utilities e trucchi a prezzi interessantissimi, scrivere a:
Mauro Di Bernardo - via G.Verdi, 1 - 33048 S.G. Al Nat. (UD)

Vendo software di vario genere per CBM 64, su nastro e su disco, a prezzi compresi tra L. 15.000 e L. 40.000

Ecco alcuni esempi di giochi su nastro da L.15.000 (su disco L.20.000) cad.
Pole Position, Dig Dug, Buck Rogers, Zaxxon, Congo Bongo, Q.Bert. Vendo inoltre il nuovissimo gioco del basket per CBM 64 a L.25.000 e su disco a L.30.000!
Alberto Borgini - via S. Rocco, 34 - 21013 - Gallarate (VA) - Tel. 0331/795770

Vendo programmi per C64 a prezzi eccezionali! Tratto solo zone Roma e Tivoli.
Gennaro Nicodemo - via Scalpelli, 23 - 00013 Tivoli (Roma) - Tel.0774/23950 ore serali

Cambio listati di programmi per VIC 20 inespanso. Mandate le vostre liste. Accetto inoltre adesioni per fondare un nuovo Club di "Vichinghi"
Enrico Brozzu - via Dei Villini, 60 - 09100 Cagliari - Tel. 371370

Commodore 64 cerco programmi da acquistare, scambiare su cassetta, soprattutto utility, gestionali, ingegneria civile, inviare lista con caratteristiche e prezzi.
Massimiliano Carabelli - via Australi, 15 - 00144 Roma

Cerco per VIC 20 light pen, espansione 16 K e tavoletta grafica a prezzi trattabili. Telefonare o scrivere al seguente indirizzo:
Albino Montella - via Verona, 13 - 84091 - Battipaglia (SA) - Tel. 0828/72107

Cambio programmi su cassetta per VIC 20 o su listati inoltre cambio cassette con molti giochi in LM e BASIC a chi mi invia le sue. Rispondo a tutti in qualsiasi caso.
Vito Savino - via P. Rebecchi, 29/G - 87100 Cosenza - Tel.0984/32116

Vendo oltre 200 programmi per CBM 64 a prezzi stracciati. Accetto anche cambi. Scrivere o telefonare a:
Luca Tamburro - via Mazzini, 67 - 80045 Pompei (NA) - Tel.081/8631918

Vendo SEGA SC 3.000 + stampante 4 colori SEGA SP400 + Monitor Computer compatibile prezzo concordabile. Si vendono anche divisi, scrivetemi e ci metteremo d'accordo.
Aldo Santiangelo - via S.Maria, 4 - 62024 Matelica (MC) - Tel. 07337/82344

Il Gruppo Astrofili Antares comunica la costituzione di una Sezione Computer ed Elettronica avvenuta il 1 giugno 1984. Le iscrizioni a questa sezione sono aperte a tutti gli interessati. Per ulteriori informazioni contattare la Segreteria presso Roberto Baldini, Via T. Randi, 9 - 48010 Cotignola - RA - Rinnoviamo l'invito ad associarsi a tutti gli appassionati romagnoli di Astronomia e ricordiamo agli astrofili italiani che il nostro Servizio Corrispondenza è a loro disposizione per ogni forma di collaborazione ed informazione. Per ogni informazione relativa all'attività e alle modalità di iscrizione al Gruppo Astrofili Antares contattare la segreteria.

Vendo VIC 20 3.5K usato pochissimo + 6 cassette della serie Impara il BASIC direttamente dal computer + registratore a L.280.000 intratt.
Bassam Merhi - via La Spezia, 23 - 00055 Ladispoli - 06/9913075

Vendo Game e utility per C64 da L.5.000. Mundial, Soccer, Strip Poker, Magic Desk, Ingegneria civile, Easy Script... è il resto dell'intera biblioteca Commodore 64 su disco o cassetta. Inviando L.1.000 avrete la lista dei programmi.
Amico Commodore - Tel.090/922307

Vendo programmi per C64 e VIC 20 al prezzo eccezionale di L.3.000 l'uno e vendo anche programmi per Apple e Sinclair Spectrum allo stesso prezzo.
Danilo Savoca - L.go Calatafimi, 8 - 90129 Palermo - Tel.485618

OGNI VENERDI' IN EDICOLA

PAPER soft

IL PRIMO SETTIMANALE DI SOFTWARE SU CARTA
PER IL TUO PERSONAL COMPUTER
Una pubblicazione della J.soft editrice

L. 1.000



spec. in abb. post. gr. 1170

Anno 1 - N° 8 - 3 agosto 1984



Editor di schermo
Galaxia



Dedalo 3-D
Puzzle



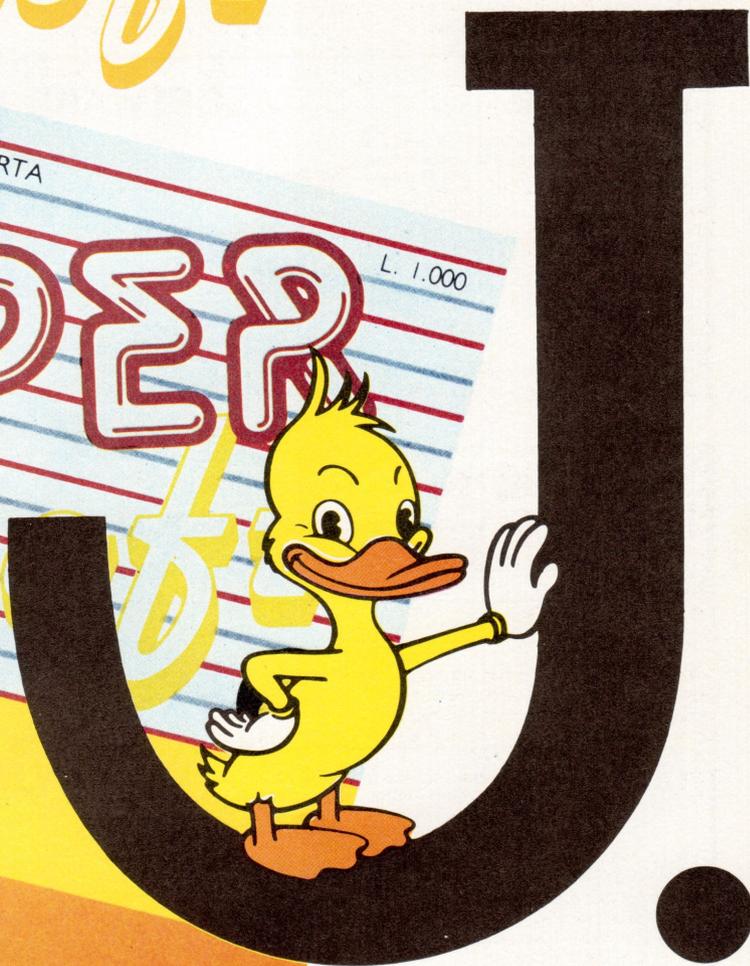
Difesa
Meteore
Carica DATA



Nevets
Poker



Nevets



PAPER SOFT,
il primo settimanale di software
per il tuo home e
personal computer
a sole 1.000 Lire.

J. soft editrice - via Rosellini, 12 - 20124 Milano
tel. 02/6888228-683797

SUPERVIC

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

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

Cod. Cod. Cod.

Cod. Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a Technoclub - Milano

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

LIBRI SOFTWARE 64 - PRACTICAL PROGRAMS FOR

THE C64 cod. BGRC007 L. 14.900
THE VIC 20 GAMES BOOK cod. BGRC004 L. 14.900

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

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

SUPERVIC

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.soft - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

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

Cod. Cod. Cod.

Cod. Cod. Cod.

Contanti allegati Assegno allegato n°

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a Technoclub - Milano

Pagherò in contrassegno al postino al ricevimento delle cassette

SOFTWARE IL GIOCO DELLA BORSA

cod. CCAMC01 L. 48.000

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

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

SUPERVIC INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soft - Via Rosellini, 12 - 20124 MILANO

COMPRO VENDO CAMBIO

VIC 20 C64 PERIF. SOFTWARE

.....

.....

.....

.....

.....

Nome Cognome

Via C.A.P.

Città Tel.

SUPERVIC è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

.....

.....

.....

.....

.....

Nome Cognome

Via C.A.P.

Città Tel.

GP550A

due in una: stampante grafica e letter quality

SEIKOSHA



La GP550A è una stampante grafica di alta qualità che offre accanto al funzionamento abituale in DATA PROCESSING MODE (stampa comune) la possibilità di stampa in NEAR LETTER QUALITY (stampa di documenti) in un'unica unità. Silenziosa, affidabile e ad un

livello di costo eccellente, si propone per una varietà di applicazioni che vanno dal data processing, alla grafica, alla produzione di documenti.

Caratteristiche:

- Stampante ad impatto a matrice di punti da 80 colonne
- Matrice di stampa 9x8 (Data Processing mode) e 9x16 (Near Letter Quality mode)
- Percorso di stampa monodirezionale (da sinistra a destra)
- Capacità grafiche con indirizzamento di 8 o 16 dot verticali
- Possibilità di ripetizione automatica di un carattere grafico
- Velocità 50 cps (Data Processing mode) e 25 cps (Near Letter Quality mode)
- Caratterizzazione: 10, 12 e 17 cpi (e relativi espansi) in DP mode; 10 e 12 cpi e relativi espansi, italico corsivo -10 cpi- e relativo espanso, super e sub scritte -17 cpi- e relativi espansi, proporzionale e relativo espanso in NLQ mode.
- Interfacce: parallela centronics (optionals Spectrum, Sinclair ZX81, seriale RS232C)
- Alimentazione carta: trattori (larghezza modulo continuo variabile da 4,5 a 10") e frizione (foglio singolo)
- Stampa 1 originale e 2 copie
- Set di 96 caratteri ASCII e 44 caratteri e simboli europei
- 8 generatori di caratteri europei a bordo
- Consumo 9W (standby) o 30W (stampa)
- Peso 5,5 KG
- Dimensioni: 305 (prof.) x 420 (largh.) x 113 (alt.) mm.
- Nastro: singolo colore su cartuccia dedicata

PERSONAL COMPUTER: le prime, le migliori!

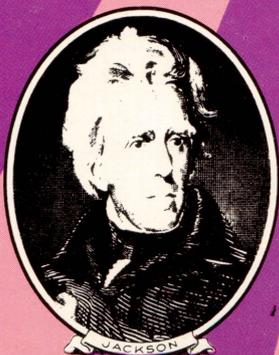
PERSONAL SOFTWARE

L'unica che presenta software
per tutti i personal:

Commodore, Apple, Sinclair, T.I., HP,
Sharp, Sega, Olivetti, ecc.



BIT: La più letta, la prima e più diffusa.
TEST: Apricot Computer
SUPERBIT - 64 pagine di programmi
per i vostri personal computer.



Con tutta la competenza del
**GRUPPO EDITORIALE
JACKSON**