

# PERSONAL SOFTWARE

ANNO 2 N. 12/13  
NOVEMBRE-DICEMBRE 1983 - L. 3.500

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON



Copiar riservata agli abbonati

- **PROGRAMMER'S TOOL KIT PER C64**
- **WEI-CH'I: UN GIOCO CINESE PER IL PET**
- **INTRODUZIONE AL LINGUAGGIO MACCHINA PER ZX81**
- **I FILE SU CASSETTA NEL T199/4A**
- **LABIRINTO DI GHIACCIO E DOT**

● **ASTERIODI PER ZX SPECTRUM**



Spedizione in abb. postale Gruppo III/70

# FLEXETTE

viaggio nella  
perfezione



seguite le vostre guide:



**RHÔNE  
POULENC  
SYSTEMES**  
settore informatico

concessionari autorizzati

**TECNODATA s.a.s.**  
di Rossolini Mauro & Dall'Olio Attilio  
Via Mazzini 12 (pall. superiore)  
41100 PARMA  
Tel. 0521 / 25 979

**PROGRAMMA UFFICIO s.a.s.**  
di Ferrero Eufemia & C.  
Corso Venezia 92/8  
10093 COLLENO (Torino)  
Tel. 011 / 41 13 585

**SDC di Berginoli Giuseppe & C. s.a.s.**  
largo Promessi Sposi 5  
20142 MILANO  
Tel. 04 35 5931 / 64 66 535

**DATEPLAN s.p.a.**  
Via Casa di Riparatio 9  
31100 BOLZANO  
Tel. 0471 / 47 721

**MIDA F.L.L.**  
Via Dietro Filippini 1/A  
37121 VERONA  
Tel. 045 / 53 05 02

**BRENUNTI MASSIMO**  
Via Piccioli 30 (uff. via Chiassi 76)  
00149 ROMA  
Tel. 06 / 41 21 665

**CSB s.a.s. di Fornasaro A. & G.**

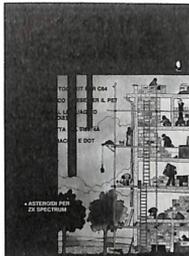
Via Via P. Scipi 2/A  
50139 FIRENZE  
Tel. 055 / 73 52 30

**TESIN & C. s.r.l.**  
Via Caravaggio 82  
30126 NAPOLI  
Tel. 081 / 64 31 22 - 64 67 59

**CPSCOM s.a.s.**  
Via Rossetiana 358  
50149 FIRENZE  
Tel. 059 / 518621

**STUDIO SINTESI**  
Via Aldighieri 6/J  
44100 FERRARA  
Tel. 059 / 43616

memorie magnetiche per computer.



In copertina: raffigurazione della costruzione di programmi in linguaggio macchina.

N. 12-13  
NOVEMBRE-DICEMBRE 1983

PERSONAL  
SOFTWARE

## ARTICOLI

- 9 CONVERSIONE DI PROGRAMMI PER ZX81 E ZX80 NUOVA ROM 3°** di *Bruno Del Medico* \_\_\_\_\_ \_ ZX81 e ZX80 nuova ROM
- 14 LABIRINTO DI GHIACCIO E DOT** di *Flavio Stella* \_\_\_\_\_ \_ C 64
- 20 TECNICHE DI INPUT E OUTPUT PER LO ZX SPECTRUM** di *Marcello Spero* \_\_\_\_\_ \_ ZX Spectrum
- 24 PROGRAMMER'S TOOL KIT PER COMMODORE 64 1°** di *Alessandro Guida* \_\_\_\_\_ \_ C 64
- 30 ASTEROIDI PER ZX SPECTRUM** di *Marcello Spero* \_\_\_\_\_ \_ ZX Spectrum
- 34 "WEI-CH'I", UN GIOCO CINESE PER PET** di *Umberto Barzaghi* \_\_\_\_\_ \_ PET/CBM
- 52 LINGUAGGIO MACCHINA PER ZX81** di *Bruno Del Medico* \_\_\_\_\_ \_ ZX81
- 64 ZX81 SUPERSCIENTIFICO** di *D. Di Mario* \_\_\_\_\_ \_ ZX81
- 66 I FILE SU CASSETTA NEL TI99/4A** di *Filippo Cerulo* \_\_\_\_\_ \_ TI99/4A
- 74 COME SALVARE CAPRA E CAVOLI NELLO ZX SPECTRUM** di *Francesco Sardo* \_\_\_\_\_ \_ ZX Spectrum

## RUBRICHE

- 5 EDITORIALE** di *Riccardo Paolillo*
- 7 POSTA**
- I SEGRETI DEI PERSONAL:**
- 78 CONTROLLO DELLA POSIZIONE DEL CURSORE** di *Italo Albanese* \_\_\_\_\_ \_ Atom-Corn
- 80 ISTRUZIONE ON ERROR GOTO N PER IL VIC 20 E C 64** di *Alessandro Guida* \_\_\_\_\_ \_ PET/CBM
- 82 RETTE AD ALTA RISOLUZIONE** di *Sergio Borsani* \_\_\_\_\_ \_ TI99/4A
- 84 VIDEOGRAFICI** di *Sergio Borsani* \_\_\_\_\_ \_ TI99/4A
- 88 LA FUNZIONE SCREENS: COME UTILIZZARLA ANCHE PER CARATTERI GRAFICI** di *Marcello Spero* \_\_\_\_\_ \_ ZX Spectrum
- CONTRIBUTI DEI LETTORI:**
- 101 EFFETTI SONORI PARTICOLARI CON LO ZX SPECTRUM** di *Mauro Carletti* \_\_\_\_\_ \_ ZX Spectrum
- 102 COME CONTROLLARE LA MEMORIA DEL TI99/4A** di *Paolo Roncali* \_\_\_\_\_ \_ TI99/4A
- 103 PICCOLI ANNUNCI**

## GUIDA

# MILANO 22-26 MAGGIO 1984

# VIDEO GAMES USA

## PERCHÈ UNA NUOVA DATA?

Per una ragione più che valida, VIDEO GAMES USA entra a far parte di BIT USA, la prestigiosa mostra di home e personal computer americani. E l'edizione '84, arricchita dalla presenza dei videogiochi, sarà più interessante che mai!

NON DIMENTICATE DUNQUE di visitare la sezione Videogiochi di BIT USA 84, dal 22 al 26 maggio, presso il Centro Commerciale Americano.



**CENTRO COMMERCIALE AMERICANO**

Via Gattamelata 5, 20149 Milano  
Tel. (02) 46.96.451 Telex 330208 USIMC-I

La mostra è realizzata in collaborazione con le riviste del **Gruppo Editoriale Jackson**.



## Gente comune

Riccardo Paolillo

A volte, ho l'impressione che il fenomeno dell'informatica personale sia sottovalutato da alcuni dei mass-media tradizionali, almeno giudicando da come vengono spesso affrontate le varie problematiche.

Non fraintendetemi: non voglio assolutamente dar vita a una polemica, che sarebbe sicuramente fuori luogo, dato che il solo fatto che tali argomenti vengano affrontati dai mezzi di informazione, è estremamente positivo per tutto il settore.

Semplicemente, ritengo, come molti altri, che l'avvento dell'informatica personale rappresenti una innovazione destinata ad avere ripercussioni sempre più sensibili nella società, e mi interessa osservare come affronta l'argomento chi lo segue dall'esterno.

Anzitutto, mi sembra, che si tenda a considerare molto minori di quelli che sono i dati relativi alla diffusione del personal e soprattutto delle persone che, pur non possedendolo, ne conoscono tutti i segreti.

Ecco quindi che vengono spesso pubblicati articoli di tipo introduttivo che vogliono sviscerare in poche righe le caratteristiche di un settore ormai affermato, ricorrendo a luoghi comuni e a informazioni a volte imprecise o incomplete.

Accanto a questi articoli, che pure se ben fatti hanno la loro importanza, raramente ne vengono presentati altri, adatti ai molti che già sanno cosa è un personal computer.

L'altro giorno in treno ho ascoltato quattro ragazzi liceali che discorrevano sui personal computer con preparazione e competenza, e parlavano di prove e

prezzi delle varie macchine con un entusiasmo che io, che pure sono solo di una mezza generazione più anziano, sicuramente avevo discutendo di motociclette con i miei amici qualche anno fa.

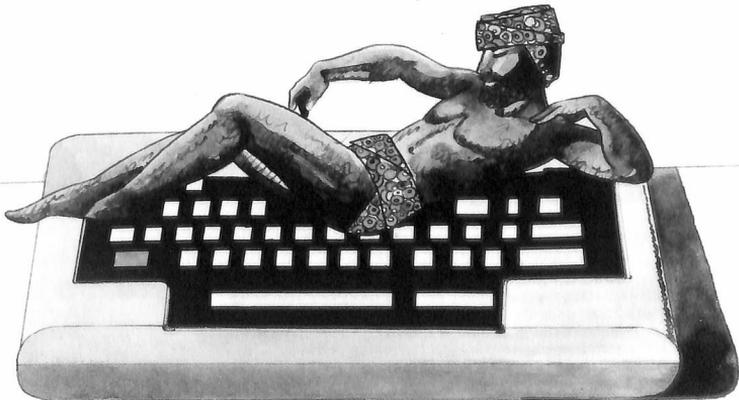
Altre volte si affronta l'argomento in chiave negativa, quando succedono fatti da cronaca nera, come le recenti intrusioni in banche dati americane realizzate mediante personal (argomento ripreso anche nel recente e discusso film *Wargames*, che mi è sembrato interessante e ben realizzato).

Ma ciò che più è sconcertante ed emblematico, è l'identikit dell'appassionato di micro informatica che ci viene spesso proposto: un personaggio abbastanza improbabile, un po' genio e un po' matto, da guardare con rispetto, ma sotto sotto da non imitare.

Su questo non sono d'accordo. Questa figura di utilizzazione della prima ora, ha lasciato il posto a persone normalissime, che utilizzano quotidianamente il proprio personal come fanno con tutti gli strumenti che la tecnologia offre per migliorare la qualità della vita.

Che l'informatica personale sia un fenomeno di massa è significativamente dimostrato anche dalla legge del mercato, che non si lascia certo influenzare dai pregiudizi: attualmente, in una città come Milano, (ma altrove il discorso è altrettanto valido, se non di più), i punti di vendita e dimostrazione di calcolatori personali, tra computer shop, grandi magazzini e catene di negozi specializzati, si contano in varie decine.

Ci sarebbe, se mai, da discutere se è normale che ormai sia più facile acquistare un computer che trovare una farmacia di turno o un idraulico disponibile, ma questo è un po' il discorso opposto e magari lo affronteremo in un'altra occasione! ■



# INIZIARE NEL MODO MIGLIORE

## Guida al SINCLAIR ZX81 ZX80-Nuova ROM

Pagg. 262  
Cod. 318B

L. 16.500

### IL LIBRO

Questa guida, con chiarezza, semplicità espositiva e ricchezza di esemplificazioni, risulta un vero e proprio strumento operativo per tutti coloro che vogliono avvicinarsi all'informatica in generale, e imparare la programmazione in BASIC, in particolare travalicando i tre calcolatori (ZX81, ZX80, ZX80 nuova ROM) a cui fa riferimento. Partendo da quello che è un computer, il lettore impara nei primi sei capitoli a programmare in BASIC, spingendosi, per chi lo vuole, oltre, sino alla programmazione in linguaggio macchina. L'ultimo capitolo riporta parecchi programmi e per ciascuno, vengono fornite, dove possibile, le diverse versioni. Tra l'altro si parlerà di file e di animazione delle figure. Per finire ben otto Appendici, essenziali ed utilissime, tra cui spiccano per interesse le due dedicate ai sistemi operativi dello ZX80, ZX80 nuova ROM e ZX81.

### SOMMARIO

Introduzione - Il calcolatore - Installazione del calcolatore  
- La programmazione - Il linguaggio BASIC - Come operare - Utilizzo della memoria - Linguaggio macchina - Esempi di programmi - caratteri del sistema - variabili del sistema - scheda BASIC ZX80 - scheda BASIC ZX80 nuova ROM e ZX81 - errori segnalati dalla macchina - sistema operativo dello ZX80 - sistema operativo dello ZX81 e nuova ROM

**GUIDA  
AL SINCLAIR  
ZX81  
ZX80 e NUOVA ROM**

**ZX81**

GRUPPO  
EDITORIALE  
JACKSON

RITA BONELLI

GRUPPO  
EDITORIALE  
JACKSON

SCONTO 20% AGLI ABBONATI  
FINO AL 28-2-'84

Per ordinare il volume  
utilizzare l'apposito tagliando  
inserito in fondo alla rivista.



**GRUPPO EDITORIALE  
JACKSON  
Divisione Libri**



OSTA

## Microdrive Spectrum

Sono un appassionato di informatica, abbonato alla Vs. rivista, particolarmente affascinato da micro, personal ed home computer.

Per questo motivo sono un estimatore, oltre che utilizzatore, dell'home computer Spectrum della Sinclair.

Prima però di acquistarlo definitivamente, accoppiandolo al mio attuale sistema personal, in modo da poterne capire le innovazioni tecnologiche e soprattutto il valore come rapporto costo prestazioni, vorrei maggiori delucidazioni sulla tanto discussa unità pseudo-drive.

Ho atteso invano articoli e test sul tema, sulle caratteristiche hardware e software (gestione dei files), sulle problematiche aperte da questo accessorio della Sinclair.

Fino ad oggi silenzio assoluto, anche se ormai da mesi il microdrive è commercializzato diffusamente.

Avete qualcosa in cantiere?

Massimo Zazzeri  
Pontassieve (FI)

*Il microdrive Sinclair non è ancora commercializzato. Sarà presumibilmente disponibile in Italia il prossimo mese di Febbraio. Alla fine dello scorso mese di Ottobre, è stato presentato alla stampa da Rebit, la divisione della GBC che commercializza i prodotti Sinclair.*

*Il microdrive è in realtà un sistema alimentato da cassette a nastro magnetico, dotato di una gestione particolarmente sofisticata. Ogni cassetta contiene un nastro alto circa un millimetro e mezzo e lungo 5 metri, capace di memorizzare oltre 85 Kbyte. La novità rispetto ai registratori normali deriva dal fatto che, essendo il nastro formato in settori e a scorrimento senza fine, si ottengono prestazioni veramente notevoli e paragonabili a quelle dei più costosi sistemi a floppy disk. Il funzionamento del microdrive è gestito completamente*

*dal sistema operativo dello Spectrum.*

*Per quanto ci riguarda, non appena il tanto sospirato microdrive sarà effettivamente disponibile, pubblicheremo dei programmi che ne prevedano l'uso.*

## Telescriventi personali

Vi scrivo per sottoporvi il seguente problema: si può con uno ZX Spectrum (48 Kbyte), e adatto hardware, fare una stazione di ricezione delle telescriventi? E se la vostra risposta è sì, sareste in grado di fornirmi il programma (in linguaggio macchina suppongo) adatto a tale scopo?

In ogni caso vi pregherei di indicarmi qualche testo su cui attingere le informazioni necessarie.

Antonio Ruggiero

*Sicuramente è possibile utilizzare lo Spectrum, come qualunque personal, anche per ricevere telex. Quello che invece è meno immediato per noi, è fornire ciò che ci richiede, in quanto dovremmo trasformarci in una software house!*

*Scherzi a parte, non abbiamo disponibili programmi per applicazioni così specifiche e possiamo soltanto invitare chi avesse affrontato problemi di questo tipo, a mettersi in contatto con la redazione.*

## Musica con Apple II

Sono in possesso di un Apple II 64 Kbyte. Oltre all'informatica (studente all'Università di Pisa), coltivo l'interesse della musica. Vorrei sapere, se possibile, alcune cose riguardo l'uso del computer come una tastiera elettronica:

- a) vorrei sapere se la mio Apple posso attaccare qualsiasi tipo di synth (oppure solo digitale o eventualmente di alcune marche);
- b) di quale interfaccia avrei bisogno per un simile collegamento;
- c) se esistono pubblicazioni su questo argomento, oppure se sono usciti articoli sulle vostre riviste.

Gabriele Paoleschi  
Pistoia

*Esistono vari sistemi per la sintesi musicale, da utilizzare con il versatissimo Apple II. Uno dei migliori è*

*forse l'unico ad essere importato in Italia è il Music System della Mountain Computer, una ditta che ha realizzato moltissime periferiche per Apple. Il sistema è costituito da 2 schede da inserire in altrettanti slot di Apple II, da una penna ottica luminosa oltre che dal relativo software su dischetto. Le prestazioni sono interessanti e prevedono la possibilità di comporre e editare su video brani di musica mediante le paddle o la penna, definire il suono dei 16 oscillatori, legare insieme in vari modi diverse sotto-composizioni e addirittura stampare su carta le diverse composizioni. Inoltre è disponibile una tastiera compatibile con il Music System. Sul n. 31 della rivista Bit è stato pubblicato un articolo dedicato al Music System, a cui può fare riferimento per maggiori approfondimenti.*

## Caratteri grafici Spectrum

Da poco possiedo uno Spectrum ed usando la cassetta in dotazione ho notato che, oltre ai segni grafici presenti nella mappa di memoria pubblicata nel manuale, vi sono altri caratteri grafici non presenti in essa. Ho provato in tutti i modi per ottenere questi caratteri, ma senza alcun successo. Quindi vi chiedo, (disperatamente), una risposta a questo mio enigma, che ormai mi tormenta da mesi.

Pietro Cordetta  
Mattola (TA)

*Raccogliamo senza indugio il suo SOS sperando di risolvere i suoi grossi problemi. Lo Spectrum oltre a un piccolo set di caratteri grafici predefiniti accessibile tramite i tasti da 1 a 8, consente all'utente di definirsi i propri caratteri nell'ambito dei vari programmi. Tramite l'istruzione USR con argomento stringa, è possibile associare ai tasti che contraddistinguono le lettere dell'alfabeto qualunque carattere grafico. Il metodo da utilizzare è spiegato in maniera abbastanza esauriente in un paragrafo del manuale d'uso, dedicato appunto alla definizione dei caratteri grafici. Da questo discorso, avrà intuito che i caratteri grafici presenti nel programma della cassetta in dotazione sono stati definiti con la USR.*

PERSONAL SOFTWARE

# Conversione di programmi per ZX81 e ZX80 nuova ROM

— terza parte —

## Ultima puntata dedicata soprattutto alle istruzioni READ, DATA e RESTORE

di Bruno Del Medico

### Le istruzioni READ, DATA e RESTORE

L'assenza di queste istruzioni costituisce sovente un grosso scoglio alla programmazione sullo ZX80. In questo articolo ho affrontato la conversione di due programmi contenenti DATA, pubblicati sul numero 2 di Personal Software a cui si rimanda.

Si tratta del *Gioco del NIM* (scritto per il PET) e del gioco *Ippica* (scritto per l'Apple II), il primo contenente DATA numerici, il secondo DATA alfabetici.

Si farà particolare attenzione al metodo di conversione delle istruzioni READ, DATA e RESTORE.

### Il Gioco del NIM: conversione di DATA numerici

L'articolo originale spiega lo svolgimento del gioco e delle principali routine e subroutine.

Le linee DATA sono la 5000 e la 5010:

```
5000 DATA 0000,0001,0010,0011,
0100,0101,0110,0111
5010 DATA 1000,1001,1010,1011,
1100,1101,1110,1111
```

```
20 SLOW
30 GOSUB 9500
35 PRINT AT 10,0;"UUDI LE REGO
40 IF INKEY="" THEN PAUSE 200
42 IF INKEY="" THEN GOSUB 96
10110111000100110101011100110
11101110001
80 LET UF=0
84 LET UI=0
88 LET ABC=1
90 DIM R(3)
100 FOR K=1 TO 3
105 FOR A(1) TO (RAND*9)+1
150 NEXT K
152 GOSUB 9000
155 GOSUB 1000
160 IF ABC=1 THEN GOTO 250
200 PRINT AT 7,2;"THE... AT 9,22;
250 IF INKEY="" THEN PAUSE 200
252 LET CH=CH+1
260 GOSUB 9000
260 PRINT AT 7,22;"TE... ",TAB 2
262 INPUT RS
320 IF RS(1) < 2 THEN GOTO 320
322 IF CODE RS(2) OR CODE RS(3)
324 IF CODE RS(2) OR CODE RS(3)
326 THEN GOTO 320
330 LET R=VAL RS(1)
332 IF R(1)-VAL RS(2) < 0 THEN GO
TO 350
340 LET A(R)=R(1)-VAL RS(2)
342 LET CH=CH+1
345 LET ABC=0
350 IF H=0 THEN LET UT=UT+1
352 IF H=0 THEN LET UI=UI+1
354 IF H=0 THEN GOTO 500
360 GOSUB 1000
360 GOTO 200
362 LET ABC=ABC+1
365 IF ABC=2 THEN LET ABC=1
370 LET CH=CH+1
370 FOR K=1 TO 3
372 FOR I=1 TO 90
375 PRINT AT K,U,CHR$ 120;
380 NEXT U
382 NEXT K
390 NEXT I THEN PRINT AT 11,20;
400 IF G=0 THEN PRINT AT 11,20;
410 IF G=0 THEN PRINT AT 11,20;
420 IF G=0 THEN PRINT AT 11,20;
430 PRINT AT 18,23;"PREMI";TAB
23;"NOME:"
500 INPUT US
502 GOTO 100
504 GOTO 100
506 FOR K=1 TO 3
508 FOR US=1 TO 19 STEP 2
1035 PRINT AT Z,U;" ";TAB U, " ";
1036 IF US(1) < 2 THEN PRINT AT 1,
1037;"TAB 1";TAB U, " ";TAB
U,US
1040 NEXT U
1042 NEXT Z
1045 PRINT AT 14,27;CH
1050 RETURN
2000 DIM D(4,4)
2002 DIM B(4,4)
2004 LET F=0
2006 FOR K=1 TO 3
2011 IF B(K)=0 THEN LET BS(K)=""
2012 IF B(K)=0 THEN GOTO 2016
2015 LET BS(K)=B(K)+3 TO 6 (K)
2018 NEXT K
2020 FOR K=1 TO 4
2022 LET D(K)=0
2030 LET D(K)=""
2032 IF D(K)="" THEN GOTO 2070
2034 LET U=1
2036 LET CS=US+3 TO U+4
2038 IF D(K)="" THEN GOTO 2070
2040 LET U=U+1
2042 LET CS=US+3 TO U+4
2044 IF D(K)="" THEN GOTO 2070
2046 LET U=U+1
2048 LET CS=US+3 TO U+4
2050 IF D(K)="" THEN GOTO 2070
2052 LET U=U+1
2054 LET CS=US+3 TO U+4
2056 IF D(K)="" THEN GOTO 2070
2058 LET U=U+1
2060 LET CS=US+3 TO U+4
2062 IF D(K)="" THEN GOTO 2070
2064 LET U=U+1
2066 LET CS=US+3 TO U+4
2068 IF D(K)="" THEN GOTO 2070
2070 LET U=U+1
2072 LET CS=US+3 TO U+4
2074 IF D(K)="" THEN GOTO 2070
2076 LET U=U+1
2078 LET CS=US+3 TO U+4
2080 IF D(K)="" THEN GOTO 2070
2082 LET U=U+1
2084 LET CS=US+3 TO U+4
2086 IF D(K)="" THEN GOTO 2070
2088 LET U=U+1
2090 LET CS=US+3 TO U+4
2092 IF D(K)="" THEN GOTO 2070
2094 LET U=U+1
2096 LET CS=US+3 TO U+4
2098 IF D(K)="" THEN GOTO 2070
2100 LET U=U+1
2102 LET CS=US+3 TO U+4
2104 IF D(K)="" THEN GOTO 2070
2106 LET U=U+1
2108 LET CS=US+3 TO U+4
2110 IF D(K)="" THEN GOTO 2070
2112 LET U=U+1
2114 LET CS=US+3 TO U+4
2116 IF D(K)="" THEN GOTO 2070
2118 LET U=U+1
2120 LET CS=US+3 TO U+4
2122 IF D(K)="" THEN GOTO 2070
2124 LET U=U+1
2126 LET CS=US+3 TO U+4
2128 IF D(K)="" THEN GOTO 2070
2130 LET U=U+1
2132 LET CS=US+3 TO U+4
2134 IF D(K)="" THEN GOTO 2070
2136 LET U=U+1
2138 LET CS=US+3 TO U+4
2140 IF D(K)="" THEN GOTO 2070
2142 LET U=U+1
2144 LET CS=US+3 TO U+4
2146 IF D(K)="" THEN GOTO 2070
2148 LET U=U+1
2150 LET CS=US+3 TO U+4
2152 IF D(K)="" THEN GOTO 2070
2154 LET U=U+1
2156 LET CS=US+3 TO U+4
2158 IF D(K)="" THEN GOTO 2070
2160 LET U=U+1
2162 LET CS=US+3 TO U+4
2164 IF D(K)="" THEN GOTO 2070
2166 LET U=U+1
2168 LET CS=US+3 TO U+4
2170 IF D(K)="" THEN GOTO 2070
2172 LET U=U+1
2174 LET CS=US+3 TO U+4
2176 IF D(K)="" THEN GOTO 2070
2178 LET U=U+1
2180 LET CS=US+3 TO U+4
2182 IF D(K)="" THEN GOTO 2070
2184 LET U=U+1
2186 LET CS=US+3 TO U+4
2188 IF D(K)="" THEN GOTO 2070
2190 LET U=U+1
2192 LET CS=US+3 TO U+4
2194 IF D(K)="" THEN GOTO 2070
2196 LET U=U+1
2198 LET CS=US+3 TO U+4
2200 IF D(K)="" THEN GOTO 2070
2202 LET U=U+1
2204 LET CS=US+3 TO U+4
2206 IF D(K)="" THEN GOTO 2070
2208 LET U=U+1
2210 LET CS=US+3 TO U+4
2212 IF D(K)="" THEN GOTO 2070
2214 LET U=U+1
2216 LET CS=US+3 TO U+4
2218 IF D(K)="" THEN GOTO 2070
2220 LET U=U+1
2222 LET CS=US+3 TO U+4
2224 IF D(K)="" THEN GOTO 2070
2226 LET U=U+1
2228 LET CS=US+3 TO U+4
2230 IF D(K)="" THEN GOTO 2070
2232 LET U=U+1
2234 LET CS=US+3 TO U+4
2236 IF D(K)="" THEN GOTO 2070
2238 LET U=U+1
2240 LET CS=US+3 TO U+4
2242 IF D(K)="" THEN GOTO 2070
2244 LET U=U+1
2246 LET CS=US+3 TO U+4
2248 IF D(K)="" THEN GOTO 2070
2250 LET U=U+1
2252 LET CS=US+3 TO U+4
2254 IF D(K)="" THEN GOTO 2070
2256 LET U=U+1
2258 LET CS=US+3 TO U+4
2260 IF D(K)="" THEN GOTO 2070
2262 LET U=U+1
2264 LET CS=US+3 TO U+4
2266 IF D(K)="" THEN GOTO 2070
2268 LET U=U+1
2270 LET CS=US+3 TO U+4
2272 IF D(K)="" THEN GOTO 2070
2274 LET U=U+1
2276 LET CS=US+3 TO U+4
2278 IF D(K)="" THEN GOTO 2070
2280 LET U=U+1
2282 LET CS=US+3 TO U+4
2284 IF D(K)="" THEN GOTO 2070
2286 LET U=U+1
2288 LET CS=US+3 TO U+4
2290 IF D(K)="" THEN GOTO 2070
2292 LET U=U+1
2294 LET CS=US+3 TO U+4
2296 IF D(K)="" THEN GOTO 2070
2298 LET U=U+1
2300 LET CS=US+3 TO U+4
2302 IF D(K)="" THEN GOTO 2070
2304 LET U=U+1
2306 LET CS=US+3 TO U+4
2308 IF D(K)="" THEN GOTO 2070
2310 LET U=U+1
2312 LET CS=US+3 TO U+4
2314 IF D(K)="" THEN GOTO 2070
2316 LET U=U+1
2318 LET CS=US+3 TO U+4
2320 IF D(K)="" THEN GOTO 2070
2322 LET U=U+1
2324 LET CS=US+3 TO U+4
2326 IF D(K)="" THEN GOTO 2070
2328 LET U=U+1
2330 LET CS=US+3 TO U+4
2332 IF D(K)="" THEN GOTO 2070
2334 LET U=U+1
2336 LET CS=US+3 TO U+4
2338 IF D(K)="" THEN GOTO 2070
2340 LET U=U+1
2342 LET CS=US+3 TO U+4
2344 IF D(K)="" THEN GOTO 2070
2346 LET U=U+1
2348 LET CS=US+3 TO U+4
2350 IF D(K)="" THEN GOTO 2070
2352 LET U=U+1
2354 LET CS=US+3 TO U+4
2356 IF D(K)="" THEN GOTO 2070
2358 LET U=U+1
2360 LET CS=US+3 TO U+4
2362 IF D(K)="" THEN GOTO 2070
2364 LET U=U+1
2366 LET CS=US+3 TO U+4
2368 IF D(K)="" THEN GOTO 2070
2370 LET U=U+1
2372 LET CS=US+3 TO U+4
2374 IF D(K)="" THEN GOTO 2070
2376 LET U=U+1
2378 LET CS=US+3 TO U+4
2380 IF D(K)="" THEN GOTO 2070
2382 LET U=U+1
2384 LET CS=US+3 TO U+4
2386 IF D(K)="" THEN GOTO 2070
2388 LET U=U+1
2390 LET CS=US+3 TO U+4
2392 IF D(K)="" THEN GOTO 2070
2394 LET U=U+1
2396 LET CS=US+3 TO U+4
2398 IF D(K)="" THEN GOTO 2070
2400 LET U=U+1
2402 LET CS=US+3 TO U+4
2404 IF D(K)="" THEN GOTO 2070
2406 LET U=U+1
2408 LET CS=US+3 TO U+4
2410 IF D(K)="" THEN GOTO 2070
2412 LET U=U+1
2414 LET CS=US+3 TO U+4
2416 IF D(K)="" THEN GOTO 2070
2418 LET U=U+1
2420 LET CS=US+3 TO U+4
2422 IF D(K)="" THEN GOTO 2070
2424 LET U=U+1
2426 LET CS=US+3 TO U+4
2428 IF D(K)="" THEN GOTO 2070
2430 LET U=U+1
2432 LET CS=US+3 TO U+4
2434 IF D(K)="" THEN GOTO 2070
2436 LET U=U+1
2438 LET CS=US+3 TO U+4
2440 IF D(K)="" THEN GOTO 2070
2442 LET U=U+1
2444 LET CS=US+3 TO U+4
2446 IF D(K)="" THEN GOTO 2070
2448 LET U=U+1
2450 LET CS=US+3 TO U+4
2452 IF D(K)="" THEN GOTO 2070
2454 LET U=U+1
2456 LET CS=US+3 TO U+4
2458 IF D(K)="" THEN GOTO 2070
2460 LET U=U+1
2462 LET CS=US+3 TO U+4
2464 IF D(K)="" THEN GOTO 2070
2466 LET U=U+1
2468 LET CS=US+3 TO U+4
2470 IF D(K)="" THEN GOTO 2070
2472 LET U=U+1
2474 LET CS=US+3 TO U+4
2476 IF D(K)="" THEN GOTO 2070
2478 LET U=U+1
2480 LET CS=US+3 TO U+4
2482 IF D(K)="" THEN GOTO 2070
2484 LET U=U+1
2486 LET CS=US+3 TO U+4
2488 IF D(K)="" THEN GOTO 2070
2490 LET U=U+1
2492 LET CS=US+3 TO U+4
2494 IF D(K)="" THEN GOTO 2070
2496 LET U=U+1
2498 LET CS=US+3 TO U+4
2500 IF D(K)="" THEN GOTO 2070
2502 LET U=U+1
2504 LET CS=US+3 TO U+4
2506 IF D(K)="" THEN GOTO 2070
2508 LET U=U+1
2510 LET CS=US+3 TO U+4
2512 IF D(K)="" THEN GOTO 2070
2514 LET U=U+1
2516 LET CS=US+3 TO U+4
2518 IF D(K)="" THEN GOTO 2070
2520 LET U=U+1
2522 LET CS=US+3 TO U+4
2524 IF D(K)="" THEN GOTO 2070
2526 LET U=U+1
2528 LET CS=US+3 TO U+4
2530 IF D(K)="" THEN GOTO 2070
2532 LET U=U+1
2534 LET CS=US+3 TO U+4
2536 IF D(K)="" THEN GOTO 2070
2538 LET U=U+1
2540 LET CS=US+3 TO U+4
2542 IF D(K)="" THEN GOTO 2070
2544 LET U=U+1
2546 LET CS=US+3 TO U+4
2548 IF D(K)="" THEN GOTO 2070
2550 LET U=U+1
2552 LET CS=US+3 TO U+4
2554 IF D(K)="" THEN GOTO 2070
2556 LET U=U+1
2558 LET CS=US+3 TO U+4
2560 IF D(K)="" THEN GOTO 2070
2562 LET U=U+1
2564 LET CS=US+3 TO U+4
2566 IF D(K)="" THEN GOTO 2070
2568 LET U=U+1
2570 LET CS=US+3 TO U+4
2572 IF D(K)="" THEN GOTO 2070
2574 LET U=U+1
2576 LET CS=US+3 TO U+4
2578 IF D(K)="" THEN GOTO 2070
2580 LET U=U+1
2582 LET CS=US+3 TO U+4
2584 IF D(K)="" THEN GOTO 2070
2586 LET U=U+1
2588 LET CS=US+3 TO U+4
2590 IF D(K)="" THEN GOTO 2070
2592 LET U=U+1
2594 LET CS=US+3 TO U+4
2596 IF D(K)="" THEN GOTO 2070
2598 LET U=U+1
2600 LET CS=US+3 TO U+4
2602 IF D(K)="" THEN GOTO 2070
2604 LET U=U+1
2606 LET CS=US+3 TO U+4
2608 IF D(K)="" THEN GOTO 2070
2610 LET U=U+1
2612 LET CS=US+3 TO U+4
2614 IF D(K)="" THEN GOTO 2070
2616 LET U=U+1
2618 LET CS=US+3 TO U+4
2620 IF D(K)="" THEN GOTO 2070
2622 LET U=U+1
2624 LET CS=US+3 TO U+4
2626 IF D(K)="" THEN GOTO 2070
2628 LET U=U+1
2630 LET CS=US+3 TO U+4
2632 IF D(K)="" THEN GOTO 2070
2634 LET U=U+1
2636 LET CS=US+3 TO U+4
2638 IF D(K)="" THEN GOTO 2070
2640 LET U=U+1
2642 LET CS=US+3 TO U+4
2644 IF D(K)="" THEN GOTO 2070
2646 LET U=U+1
2648 LET CS=US+3 TO U+4
2650 IF D(K)="" THEN GOTO 2070
2652 LET U=U+1
2654 LET CS=US+3 TO U+4
2656 IF D(K)="" THEN GOTO 2070
2658 LET U=U+1
2660 LET CS=US+3 TO U+4
2662 IF D(K)="" THEN GOTO 2070
2664 LET U=U+1
2666 LET CS=US+3 TO U+4
2668 IF D(K)="" THEN GOTO 2070
2670 LET U=U+1
2672 LET CS=US+3 TO U+4
2674 IF D(K)="" THEN GOTO 2070
2676 LET U=U+1
2678 LET CS=US+3 TO U+4
2680 IF D(K)="" THEN GOTO 2070
2682 LET U=U+1
2684 LET CS=US+3 TO U+4
2686 IF D(K)="" THEN GOTO 2070
2688 LET U=U+1
2690 LET CS=US+3 TO U+4
2692 IF D(K)="" THEN GOTO 2070
2694 LET U=U+1
2696 LET CS=US+3 TO U+4
2698 IF D(K)="" THEN GOTO 2070
2700 LET U=U+1
2702 LET CS=US+3 TO U+4
2704 IF D(K)="" THEN GOTO 2070
2706 LET U=U+1
2708 LET CS=US+3 TO U+4
2710 IF D(K)="" THEN GOTO 2070
2712 LET U=U+1
2714 LET CS=US+3 TO U+4
2716 IF D(K)="" THEN GOTO 2070
2718 LET U=U+1
2720 LET CS=US+3 TO U+4
2722 IF D(K)="" THEN GOTO 2070
2724 LET U=U+1
2726 LET CS=US+3 TO U+4
2728 IF D(K)="" THEN GOTO 2070
2730 LET U=U+1
2732 LET CS=US+3 TO U+4
2734 IF D(K)="" THEN GOTO 2070
2736 LET U=U+1
2738 LET CS=US+3 TO U+4
2740 IF D(K)="" THEN GOTO 2070
2742 LET U=U+1
2744 LET CS=US+3 TO U+4
2746 IF D(K)="" THEN GOTO 2070
2748 LET U=U+1
2750 LET CS=US+3 TO U+4
2752 IF D(K)="" THEN GOTO 2070
2754 LET U=U+1
2756 LET CS=US+3 TO U+4
2758 IF D(K)="" THEN GOTO 2070
2760 LET U=U+1
2762 LET CS=US+3 TO U+4
2764 IF D(K)="" THEN GOTO 2070
2766 LET U=U+1
2768 LET CS=US+3 TO U+4
2770 IF D(K)="" THEN GOTO 2070
2772 LET U=U+1
2774 LET CS=US+3 TO U+4
2776 IF D(K)="" THEN GOTO 2070
2778 LET U=U+1
2780 LET CS=US+3 TO U+4
2782 IF D(K)="" THEN GOTO 2070
2784 LET U=U+1
2786 LET CS=US+3 TO U+4
2788 IF D(K)="" THEN GOTO 2070
2790 LET U=U+1
2792 LET CS=US+3 TO U+4
2794 IF D(K)="" THEN GOTO 2070
2796 LET U=U+1
2798 LET CS=US+3 TO U+4
2800 IF D(K)="" THEN GOTO 2070
2802 LET U=U+1
2804 LET CS=US+3 TO U+4
2806 IF D(K)="" THEN GOTO 2070
2808 LET U=U+1
2810 LET CS=US+3 TO U+4
2812 IF D(K)="" THEN GOTO 2070
2814 LET U=U+1
2816 LET CS=US+3 TO U+4
2818 IF D(K)="" THEN GOTO 2070
2820 LET U=U+1
2822 LET CS=US+3 TO U+4
2824 IF D(K)="" THEN GOTO 2070
2826 LET U=U+1
2828 LET CS=US+3 TO U+4
2830 IF D(K)="" THEN GOTO 2070
2832 LET U=U+1
2834 LET CS=US+3 TO U+4
2836 IF D(K)="" THEN GOTO 2070
2838 LET U=U+1
2840 LET CS=US+3 TO U+4
2842 IF D(K)="" THEN GOTO 2070
2844 LET U=U+1
2846 LET CS=US+3 TO U+4
2848 IF D(K)="" THEN GOTO 2070
2850 LET U=U+1
2852 LET CS=US+3 TO U+4
2854 IF D(K)="" THEN GOTO 2070
2856 LET U=U+1
2858 LET CS=US+3 TO U+4
2860 IF D(K)="" THEN GOTO 2070
2862 LET U=U+1
2864 LET CS=US+3 TO U+4
2866 IF D(K)="" THEN GOTO 2070
2868 LET U=U+1
2870 LET CS=US+3 TO U+4
2872 IF D(K)="" THEN GOTO 2070
2874 LET U=U+1
2876 LET CS=US+3 TO U+4
2878 IF D(K)="" THEN GOTO 2070
2880 LET U=U+1
2882 LET CS=US+3 TO U+4
2884 IF D(K)="" THEN GOTO 2070
2886 LET U=U+1
2888 LET CS=US+3 TO U+4
2890 IF D(K)="" THEN GOTO 2070
2892 LET U=U+1
2894 LET CS=US+3 TO U+4
2896 IF D(K)="" THEN GOTO 2070
2898 LET U=U+1
2900 LET CS=US+3 TO U+4
2902 IF D(K)="" THEN GOTO 2070
2904 LET U=U+1
2906 LET CS=US+3 TO U+4
2908 IF D(K)="" THEN GOTO 2070
2910 LET U=U+1
2912 LET CS=US+3 TO U+4
2914 IF D(K)="" THEN GOTO 2070
2916 LET U=U+1
2918 LET CS=US+3 TO U+4
2920 IF D(K)="" THEN GOTO 2070
2922 LET U=U+1
2924 LET CS=US+3 TO U+4
2926 IF D(K)="" THEN GOTO 2070
2928 LET U=U+1
2930 LET CS=US+3 TO U+4
2932 IF D(K)="" THEN GOTO 2070
2934 LET U=U+1
2936 LET CS=US+3 TO U+4
2938 IF D(K)="" THEN GOTO 2070
2940 LET U=U+1
2942 LET CS=US+3 TO U+4
2944 IF D(K)="" THEN GOTO 2070
2946 LET U=U+1
2948 LET CS=US+3 TO U+4
2950 IF D(K)="" THEN GOTO 2070
2952 LET U=U+1
2954 LET CS=US+3 TO U+4
2956 IF D(K)="" THEN GOTO 2070
2958 LET U=U+1
2960 LET CS=US+3 TO U+4
2962 IF D(K)="" THEN GOTO 2070
2964 LET U=U+1
2966 LET CS=US+3 TO U+4
2968 IF D(K)="" THEN GOTO 2070
2970 LET U=U+1
2972 LET CS=US+3 TO U+4
2974 IF D(K)="" THEN GOTO 2070
2976 LET U=U+1
2978 LET CS=US+3 TO U+4
2980 IF D(K)="" THEN GOTO 2070
2982 LET U=U+1
2984 LET CS=US+3 TO U+4
2986 IF D(K)="" THEN GOTO 2070
2988 LET U=U+1
2990 LET CS=US+3 TO U+4
2992 IF D(K)="" THEN GOTO 2070
2994 LET U=U+1
2996 LET CS=US+3 TO U+4
2998 IF D(K)="" THEN GOTO 2070
3000 LET U=U+1
3002 LET CS=US+3 TO U+4
3004 IF D(K)="" THEN GOTO 2070
3006 LET U=U+1
3008 LET CS=US+3 TO U+4
3010 IF D(K)="" THEN GOTO 2070
3012 LET U=U+1
3014 LET CS=US+3 TO U+4
3016 IF D(K)="" THEN GOTO 2070
3018 LET U=U+1
3020 LET CS=US+3 TO U+4
3022 IF D(K)="" THEN GOTO 2070
3024 LET U=U+1
3026 LET CS=US+3 TO U+4
3028 IF D(K)="" THEN GOTO 2070
3030 LET U=U+1
3032 LET CS=US+3 TO U+4
3034 IF D(K)="" THEN GOTO 2070
3036 LET U=U+1
3038 LET CS=US+3 TO U+4
3040 IF D(K)="" THEN GOTO 2070
3042 LET U=U+1
3044 LET CS=US+3 TO U+4
3046 IF D(K)="" THEN GOTO 2070
3048 LET U=U+1
3050 LET CS=US+3 TO U+4
3052 IF D(K)="" THEN GOTO 2070
3054 LET U=U+1
3056 LET CS=US+3 TO U+4
3058 IF D(K)="" THEN GOTO 2070
3060 LET U=U+1
3062 LET CS=US+3 TO U+4
3064 IF D(K)="" THEN GOTO 2070
3066 LET U=U+1
3068 LET CS=US+3 TO U+4
3070 IF D(K)="" THEN GOTO 2070
3072 LET U=U+1
3074 LET CS=US+3 TO U+4
3076 IF D(K)="" THEN GOTO 2070
3078 LET U=U+1
3080 LET CS=US+3 TO U+4
3082 IF D(K)="" THEN GOTO 2070
3084 LET U=U+1
3086 LET CS=US+3 TO U+4
3088 IF D(K)="" THEN GOTO 2070
3090 LET U=U+1
3092 LET CS=US+3 TO U+4
3094 IF D(K)="" THEN GOTO 2070
3096 LET U=U+1
3098 LET CS=US+3 TO U+4
3100 IF D(K)="" THEN GOTO 2070
3102 LET U=U+1
3104 LET CS=US+3 TO U+4
3106 IF D(K)="" THEN GOTO 2070
3108 LET U=U+1
3110 LET CS=US+3 TO U+4
3112 IF D(K)="" THEN GOTO 2070
3114 LET U=U+1
3116 LET CS=US+3 TO U+4
3118 IF D(K)="" THEN GOTO 2070
3120 LET U=U+1
3122 LET CS=US+3 TO U+4
3124 IF D(K)="" THEN GOTO 2070
3126 LET U=U+1
3128 LET CS=US+3 TO U+4
3130 IF D(K)="" THEN GOTO 2070
3132 LET U=U+1
3134 LET CS=US+3 TO U+4
3136 IF D(K)="" THEN GOTO 2070
3138 LET U=U+1
3140 LET CS=US+3 TO U+4
3142 IF D(K)="" THEN GOTO 2070
3144 LET U=U+1
3146 LET CS=US+3 TO U+4
3148 IF D(K)="" THEN GOTO 2070
3150 LET U=U+1
3152 LET CS=US+3 TO U+4
3154 IF D(K)="" THEN GOTO 2070
3156 LET U=U+1
3158 LET CS=US+3 TO U+4
3160 IF D(K)="" THEN GOTO 2070
3162 LET U=U+1
3164 LET CS=US+3 TO U+4
3166 IF D(K)="" THEN GOTO 2070
3168 LET U=U+1
3170 LET CS=US+3 TO U+4
3172 IF D(K)="" THEN GOTO 2070
3174 LET U=U+1
3176 LET CS=US+3 TO U+4
3178 IF D(K)="" THEN GOTO 2070
3180 LET U=U+1
3182 LET CS=US+3 TO U+4
3184 IF D(K)="" THEN GOTO 2070
3186 LET U=U+1
3188 LET CS=US+3 TO U+4
3190 IF D(K)="" THEN GOTO 2070
3192 LET U=U+1
3194 LET CS=US+3 TO U+4
3196 IF D(K)="" THEN GOTO 2070
3198 LET U=U+1
3200 LET CS=US+3 TO U+4
3202 IF D(K)="" THEN GOTO 2070
3204 LET U=U+1
3206 LET CS=US+3 TO U+4
3208 IF D(K)="" THEN GOTO 2070
3210 LET U=U+1
3212 LET CS=US+3 TO U+4
3214 IF D(K)="" THEN GOTO 2070
3216 LET U=U+1
3218 LET CS=US+3 TO U+4
3220 IF D(K)="" THEN GOTO 2070
3222 LET U=U+1
3224 LET CS=US+3 TO U+4
3226 IF D(K)="" THEN GOTO 2070
3228 LET U=U+1
3230 LET CS=US+3 TO U+4
3232 IF D(K)="" THEN GOTO 2070
3234 LET U=U+1
3236 LET CS=US+3 TO U+4
3238 IF D(K)="" THEN GOTO 2070
3240 LET U=U+1
3242 LET CS=US+3 TO U+4
3244 IF D(K)="" THEN GOTO 2070
3246 LET U=U+1
3248 LET CS=US+3 TO U+4
3250 IF D(K)="" THEN GOTO 2070
3252 LET U=U+1
3254 LET CS=US+3 TO U+4
3256 IF D(K)="" THEN GOTO 2070
3258 LET U=U+1
3260 LET CS=US+3 TO U+4
3262 IF D(K)="" THEN GOTO 2070
3264 LET U=U+1
3266 LET CS=US+3 TO U+4
3268 IF D(K)="" THEN GOTO 2070
3270 LET U=U+1
3272 LET CS=US+3 TO U+4
3274 IF D(K)="" THEN GOTO 2070
3276 LET U=U+1
3278 LET CS=US+3 TO U+4
3280 IF D(K)="" THEN GOTO 2070
3282 LET U=U+1
3284 LET CS=US+3 TO U+4
3286 IF D(K)="" THEN GOTO 2070
3288 LET U=U+1
3290 LET CS=US+3 TO U+4
3292 IF D(K)="" THEN GOTO 2070
3294 LET U=U+1
3296 LET CS=US+3 TO U+4
3298 IF D(K)="" THEN GOTO 2070
3300 LET U=U+1
3302 LET CS=US+3 TO U+4
3304 IF D(K)="" THEN GOTO 2070
3306 LET U=U+1
3308 LET CS=US+3 TO U+4
3310 IF D(K)="" THEN GOTO 2070
3312 LET U=U+1
3314 LET CS=US+3 TO U+4
3316 IF D(K)="" THEN GOTO 2070
3318 LET U=U+1
3320 LET CS=US+3 TO U+4
3322 IF D(K)="" THEN GOTO 2070
3324 LET U=U+1
3326 LET CS=US+3 TO U+4
3328 IF D(K)="" THEN GOTO 2070
3330 LET U=U+1
3332 LET CS=US+3 TO U+4
3334 IF D(K)="" THEN GOTO 2070
3336 LET U=U+1
3338 LET CS=US+3 TO U+4
3340 IF D(K)="" THEN GOTO 2070
3342 LET U=U+1
3344 LET CS=US+3 TO U+4
3346 IF D(K)="" THEN GOTO 2070
3348 LET U=U+1
3350 LET CS=US+3 TO U+4
3352 IF D(K)="" THEN GOTO 2070
3354 LET U=U+1
3356 LET CS=US+3 TO U+4
3358 IF D(K)="" THEN GOTO 2070
3360 LET U=U+1
3362 LET CS=US+3 TO U+4
3364 IF D(K)="" THEN GOTO 2070
3366 LET U=U+1
3368 LET CS=US+3 TO U+4
3370 IF D(K)="" THEN GOTO 2070
3372 LET U=U+1
3374 LET CS=US+3 TO U+4
3376 IF D(K)="" THEN GOTO 2070
3378 LET U=U+1
3380 LET CS=US+3 TO U+4
3382 IF D(K)="" THEN GOTO 2070
3384 LET U=U+1
3386 LET CS=US+3 TO U+4
3388 IF D(K)="" THEN GOTO 2070
3390 LET U=U+1
3392 LET CS=US+3 TO U+4
3394 IF D(K)="" THEN GOTO 2070
3396 LET U=U+1
3398 LET CS=US+3 TO U+4
3400 IF D(K)="" THEN GOTO 2070
3402 LET U=U+1
3404 LET CS=US+3 TO U+4
3406 IF D(K)="" THEN GOTO 2070
3408 LET U=U+1
3410 LET CS=US+3 TO U+4
3412 IF D(K)="" THEN GOTO 2070
3414 LET U=U+1
3416 LET CS=US+3 TO U+4
3418 IF D(K)="" THEN GOTO 2070
3420 LET U=U+1
3422 LET CS=US+3 TO U+4
3424 IF D(K)="" THEN GOTO 2070
3426 LET U=U+1
3428 LET CS=US+3 TO U+4
3430 IF D(K)="" THEN GOTO 2070
3432 LET U=U+1
3434 LET CS=US+3 TO U+4
3436 IF D(K)="" THEN GOTO 2070
3438 LET U=U+1
3440 LET CS=US+3 TO U+4
3442 IF D(K)="" THEN GOTO 2070
3444 LET U=U+1
3446 LET CS=US+3 TO U+4
3448 IF D(K)="" THEN GOTO 2070
3450 LET U=U+1
3452 LET CS=US+3 TO U+4
3454 IF D(K)="" THEN GOTO 2070
3456 LET U=U+1
3458 LET CS=US+3 TO U+4
3460 IF D(K)="" THEN GOTO 2070
3462 LET U=U+1
3464 LET CS=US+3 TO U+4
3466 IF D(K)="" THEN GOTO 2070
3468 LET U=U+1
3470 LET CS=US+3 TO U+4
3472 IF D(K)="" THEN GOTO 2070
3474 LET U=U+1
3476 LET CS=US+3 TO U+4
3478 IF D(K)="" THEN GOTO 2070
3480 LET U=U+1
3482 LET CS=US+3 TO U+4
3484 IF D
```



## Conversione di programmi per ZX81 e ZX80 nuova ROM

Tutti i dati contenuti in queste linee sono numerici, e sono composti da 4 cifre ciascuno, il che facilita la conversione. Innanzitutto, si può inizializzare una stringa A\$ (se non esiste nessun'altra variabile A\$ nel programma) così composta:

```
LET A$="00000001001000110
100010101100111
1000100110101011100
110111010111"
```

così tutti i dati delle due linee 5000 e 5010 sono immagazzinati in A\$, senza alcun segno di separazione. Poiché ogni dato è composto da 4 caratteri, si può facilmente recuperare per mezzo di una scansione effettuata a passo 4 lungo la stringa.

Nel BASIC standard, il dato da leggere viene individuato dall'istruzione READ utilizzando un puntatore di dato interno al sistema.

Si deve quindi inizializzare anche per la stringa A\$ un puntatore di dato, per esempio la variabile U. Per leggere il dato 1, U deve avere valore 1 e l'istruzione relativa è:

```
LET DATO = VAL A$(U * 4-3 TO U * 4)
```

che equivale a:

```
READ DATO
```

Se si vuole leggere il tredicesimo dato della stringa A\$, l'istruzione da usare è:

```
xxa LET U = 13
xxb LET DATO = VAL A$(U * 4-3 TO U * 4)
xxc LET U = 1
```

che equivale a:

```
xxa FOR U = 0 TO 12
xxb READ DATO
xxc NEXT U
xxc RESTORE
```

Ovviamente, se i dati immagazzinati nella stringa A\$ sono composti da 6 caratteri, l'istruzione per leggere il tredicesimo dato diventa:

```
xxb LET DATO = VAL A$(U * 6-5 TO U * 6)
```

Spesso i dati numerici sono composti da una quantità diversa di cifre; per esempio in un programma si può trovare una linea come questa:

```
9000 DATA 1, 16509, 0, 26
```

Esiste un metodo semplicissimo

per uniformare la quantità di caratteri in ogni dato numerico, e consiste nell'anteporre ai dati più corti degli zeri. La stringa A\$ relativa alla linea 9000 diventa:

```
9000 LET A$ =
"00001165090000000026"
```

ed il quinto dato di A\$ può essere letto così:

```
9005 LET U = 5
9010 LET DATO = VAL A$(U * 5-4 TO U * 5).
```

```
3 REM PAGLIETTE
20 GOSUB 3000
22 PRINT "O SONO IL TUO AVER
SARIO IN QUESTO GIOCO?"
24 PRINT "SULLO SCHERMO APPARE
UN CERTO NUMERO DI PAGLIETTE".
25 PRINT "TU OGGI DOBBIAMO TOG
LIERNE 1,2 OTRE ALLA VOLTA".
34 PRINT "CHI TOGLIE L ULTIMA
95 PRINT AT 21,0;"PER COMINCIA
RE AVER NEULINE".
38 INPUT US
CLS
95 LET P1=0
105 LET P2=0
100 GOSUB 3000
110 RAND
120 LET PAG=INT (RAND*16)+15
130 LET G=0
140 GOSUB 3000
155 PRINT AT 13,0;"
156 PRINT AT 10,0;"VINTE IO ";P
1740 LET U=ANTE IO *4+P2
157 PRINT AT 20,0;"
190 PRINT AT 13,0;"CI SONO ";PA
200 PRINT AT 16,0;"
202 PRINT AT 16,0;"QUANTE NE TO
217 7 (1-3)
210 INPUT N5
212 LET G=1
215 IF LEN N5<1 THEN GOTO 210
220 IF CODE N5<0 OR CODE N5>31
THEN GOTO 210
230 LET NEULINE N5.
231 PRINT AT 16,0;"
240 PRINT AT 15,0;"NE HAI TOLTE
245 LET PAG=PAG-NUM
250 GOSUB 3000
260 PRINT AT 13,0;"CI SONO ";PA
31. PAGLIETTE
320 IF PAG=1 THEN GOTO 700
330 PRINT AT 16,0;"ATTENTO TOCC
A R ME PAGLIETTE NEULINE
340 INPUT US
341 LET G=0
350 GOSUB 3000
390 PRINT AT 17,0;"
391 PRINT AT 15,0;"
396 PRINT AT 15,0;"ME HO LEVATE
397 TOG
486 GOSUB 3000
480 PRINT AT 13,0;"CI SONO ";PA
G. PAGLIETTE
486 LET G=0 TOG
470 IF PAG=1 THEN GOTO 500
480 GOTO 300
700 PRINT AT 15,0;"
710 PRINT AT 15,0;"
720 GOTO 900
880 PRINT AT 15,0;"
810 PRINT AT 15,0;"HAI U
INTO TU
811 LET G=0+1
900 PRINT AT 15,0;"VINTE IO ";P
11740 LET U=ANTE TU *4+P2
510 PRINT AT 21,0;"UOGI CONTINU
30E 7 (5-7)
920 IF INKEY="" THEN PRUSE 200
38
930 IF INKEY="" THEN GOTO 1300
940 GOSUB 3000
945 PRINT AT 10,16;"CABO"TAB 1
31. SPERO CHE TI SIA 7/06 167D10
54710
950 STOP
9500 FOR V=0 TO PAG-1
9510 FOR Y=0 TO 11
9520 PRINT AT Y,X;"
9530 NEXT Y
9540 NEXT X
9550 RETURN
9560 IF S=1 THEN FOR C=G+PAG+NUM
-1 TO G:PAG STEP
9570 FOR L=10 TO 6 STEP -1
9580 PRINT AT L,C;"
9590 NEXT C
9600 NEXT X
7000 LET Z=PAG-4:INT (PAG/4)
7100 LET TOCC=INT (INT (1+3)/4)
7200 IF Z=1 THEN LET TOG=INT (R)
0+1
7300 LET PAG=PAG-TOG
7400 RETURN
9610 PRINT "
9620 PRINT "
9630 PRINT "
9640 PRINT "
9650 PRINT "
9660 RETURN
```

Listato 2. *Pagliette. Una sfida contro il computer. Da una fila di pagliette bisogna toglierne 1, 2 o 3 a turno. Chi toglie l'ultima perde. Il computer è velocissimo (muove in meno di un secondo) e non gioca affatto male. Questo programma gira su ZX81 e ZX80/8 Kbyte e richiede 4 Kbyte di memoria RAM.*

## Conversione di programmi per ZX81 e ZX80 nuova ROM

Una routine abbastanza comune a molti programmi è quella che assegna il valore dei dati contenuti nella linea DATA, in successione agli elementi di un vettore.

Per esempio, una routine che assegni ai quattro elementi di un vettore N(4) il valore dei quattro dati contenuti nella stringa A\$ della linea 9000 è la seguente:

```
9000 DIM N(4)
9010 FOR U = 1 TO 4
9015 LET N(U) = VAL A$(U ★
5-4 TO U ★ 5)
9020 NEXT U
```

Si riconsideri il programma *Gioco del NIM*. I dati contenuti nella stringa A\$ vengono letti in due occasioni. La prima volta vengono letti dalla routine 2010/2015 del listato originale, o 2010/2016 del listato 3:

```
2010 FOR K = 1 TO 3
FOR J = 0 TO B(K)
READ B$(K)
NEXT J
RESTORE
NEXT K
```

B(K) è un vettore composto da 3 elementi, ed infatti il ciclo FOR K va da 1 a 3. Se si assegnano i seguenti valori agli elementi del vettore B:

B(1) = 1    B(2) = 0    B(3) = 2

si ottiene il seguente svolgimento del ciclo:

Ciclo esterno	Ciclo interno
K = 1    B(K) = 1	J = 0 B\$(1) = "0000" J = 1 B\$(1) = "0001"
K = 2    B(K) = 0	J = 0 B\$(2) = "0000" J = 0 B\$(2) = "0000"
K = 3    B(K) = 2	J = 0 B\$(3) = "0000" J = 1 B\$(3) = "0001" J = 2 B\$(3) = "0010"

Al termine della routine i tre elementi di B\$ valgono rispettivamente:

B\$(1) = "0001"    B\$(2) = "0000"  
B\$(3) = "0010"

cioè nella stringa B\$(x) è sempre contenuto l'equivalente binario di B(x).

Nella versione per ZX81 si hanno delle complicazioni. Il ciclo FOR J, che va da 0 a B(K), infatti, può essere ommesso in quanto l'operazione di slicing effettuata su A\$ contiene già l'indicazione che l'elemento da selezionare è il B(K) e non occorre leggere tutti gli elementi in sequenza per posizionare il puntatore di dato interno al sistema.

```
2010 FOR K = 1 TO 3
2015 LET B$(K) = A$(B(K) ★ 4-3
TO B(K) ★ 4)
2016 NEXT K
```

Si pensi come prima allo svolgimento del ciclo per vedere che cosa non funziona, assumendo per il vettore B i medesimi valori precedenti: 1, 0 e 2.

Ciclo

K = 1    B(K) = 1    B\$(1) = A\$(1 ★ 4-3 TO 1 ★ 4)

cioè: "0000"

K = 2    B(K) = 0    B\$(K) = A\$(0 ★ 4-3 TO 0 ★ 4)

cioè: ?

Alla prima esecuzione del ciclo FOR K, la stringa B\$(1) che dovrebbe essere uguale a "0001" è invece uguale a "0000"; questo perché il ciclo FOR J andava da 0 a 4, e quindi contava una posizione in più.

Si può rimediare spostando il dato 0000 *alla fine* della stringa A\$, come nel listato 3.

Quando K è uguale a 2 e B(K) è uguale a 0, si ottiene un messaggio di errore perché il numero di caratteri di una stringa è sempre positivo, mentre con l'istruzione:

B\$ = A\$(0 ★ 4-3 TO 0 ★ 4)

si puntano elementi inesistenti di A\$. Inoltre, anche se la precedente istruzione leggesse un dato, si tratterebbe di un dato errato perché abbiamo spostato 0000 in fondo alla stringa A\$.

Si può ovviare a ciò inserendo le due linee 2011 e 2012 (listato 1).

Nel secondo caso vengono letti i dati della stringa A\$ (linea 2040), e tutto funziona regolarmente. Si inizializzi un puntatore di dato U (linea 2035) che viene incrementato di uno ad ogni lettura (linea 2062) e viene rimesso a valore 1 in coincidenza con l'istruzione RESTORE del listato originale.

### Altri commenti alla conversione del Gioco del NIM

Il computer tiene il conteggio delle partite giocate (variabile CP) delle proprie vittorie (VI) e di quelle dello sfidante (VT).

La prima mossa viene concessa alternativamente a uno o all'altro dei giocatori tramite la variabile ABC. Sono state introdotte numerose linee di verifica dell'input non previste nel listato originale, perché lo ZX81 si blocca se riceve un input alfabetico quando ne aspetta uno numerico.

Tutti i vettori vengono *dichiarati* prima di essere usati; la linea

2075 S = S + S(J)

non può essere tradotta semplicemente:

2075 LET S = S + S(J)

se non abbiamo dichiarato in precedenza:

2000 DIM S(4)

Questo dimensionamento iniziale è superfluo in molti computer, ma nello ZX81 è sempre indispensabile.

La linea 2080 rimanda alla 2090



# OGGI ANTICIPAZIONI SUL FUTURO

**SIOA**  
salone  
dell'informatica,  
della telematica  
e della  
organizzazione  
aziendale

Bologna,  
25-29 febbraio 1984  
quartiere fieristico

Informatica  
Telecomunicazioni  
Telematica  
Servizi di consulenza  
ed assistenza  
alle imprese  
Attrezzature  
per l'ufficio

promosso da:

ANIE - Associazione Nazionale  
Industrie Elettrotecniche  
ed Elettroniche

Ente Autonomo  
per le Fiere di Bologna

Fondazione Guglielmo Marconi

IRSAC - Istituto di Studi  
e Ricerche sulle Attività  
Commerciali e Produttive

Gestione:

G.E.M.A. General Management S.r.l.  
Direzione Operativa:  
Bologna, via de' Buttieri, 7/2A  
Tel. 051 / 308952 - 340882  
Telex 510878



## Conversione di programmi per ZX81 e ZX80 nuova ROM

gono utilizzate tutte le 24 linee dello schermo, con la routine 620/830.

Quando lo schermo viene allungato con l'istruzione POKE 16418,0 non può essere eseguita l'istruzione INPUT per cui, volendo continuare il programma, bisogna usare l'istruzione POKE 16418,2 che riserva nuovamente al computer le due linee in fondo.

La linea:

800 PAUSE 20000

ferma il programma per più di sei minuti, ma in pratica basta premere qualsiasi tasto ed il programma riprende. Infine si osservi che PAUSE 20000 può essere considerata equivalente a:

800 IF INKEY\$ = " " THEN GOTOG 800. ■

```

1 REM 4/A: VERSIONE STANDARD
10 DIM N$(quantità di dati meno uno)
15 FOR I = 0 TO quantità di dati meno uno
20 READ N$(I)
25 NEXT
30 DATA X,XXX,XX,X
  
```

```

1 REM 4/B: VERSIONE PER ZX81
10 LET D$ = "X,XXX,XX,X,"
15 LET H18 = 0
20 LET J = 1
30 DIM N$(quantità di dati, numero di caratteri del dato più lungo)
35 FOR I = 1 TO quantità di dati
40 FOR J = J TO LEN D$
45 IF D$(J) = " " THEN LET H18 = H18 + 1
50 IF H18 = I THEN GOTO 60
55 NEXT J
60 FOR H = J + 1 TO LEN D$
65 IF D$(H) = " " THEN GOTO 70
70 NEXT H
75 LET N$(I) = D$(J + 1 TO H-1)
80 LET J = J + 1
85 NEXT I
  
```

Listato 4. Routine generalizzata per il trasferimento di DATA in un vettore. Al termine delle due routine gli elementi di N\$ contengono i dati della linea 30 o 10 rispettivamente.

La stringa D\$ deve contenere una virgola all'inizio ed una alla fine, oltre che una tra ogni singolo dato. La versione per ZX81 può essere usata con dati alfabetici o numerici: nel secondo caso gli elementi di N\$ vengono trattati con la funzione VAL.

# ZX81



Se volete riceverlo velocemente compilate e spedite in busta il "Coupon Sinclair" e riceverete in OMAGGIO il famoso libro "Guida al Sinclair ZX81" di ben 264 pagine, del valore di L. 16.500.

## EXELCO

Via G. Verdi, 23/25  
20095 - CUSANO MILANINO (MILANO)

Descrizione	Qt.	Prezzo unitario	Totale L.
Personal Computer ZX81, con alimentatore 0,7 A, completo di manuale originale Inglese e cavetti di collegamento al televisore e registratore.		L. 99.000	
Modulo di espansione di memoria 16K RAM		L. 99.000	
Modulo di espansione di memoria 32K RAM		L. 160.000	
Modulo di espansione di memoria 64K RAM		L. 250.000	
Interfaccia Centronics		L. 120.000	
Espansione Grafica		L. 130.000	
Stampante ZX Printer		L. 180.000	
Cavo coll. interfaccia Centronics		L. 38.000	

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

Nome

Cognome

Via

Città

Data     C.A.P.

Desidero ricevere la fattura  SI  NO

Partita I.V.A. o, per i privati

Codice Fiscale

Acconto L.



Sarà data precedenza alle spedizioni, se assieme all'ordine verrà incluso un anticipo di almeno L. 10.000. Aggiungere L. 5.000 per il recapito a domicilio. I prezzi vanno maggiorati dell'I.V.A. 18% e sono validi fino a giugno 1984.







**IL TUO PRIMO COMPUTER**



**ZX81**  
CON ALIMENTATORE

**REBIT**  
COMPUTER  
A DIVISION OF G.B.C.

**sinclair**

**Il computer più venduto nel mondo**

**£. 99.000**

Il prezzo non è comprensivo di IVA

# Io oggi ho scelto MPF II E sono soddisfatto.



*MPF II l'utilizzo dappertutto. È leggero, compatto, grande come una agenda. Con lui oggi muovo i primi passi nell'affascinante mondo dell'informatica. Sono sicuro che insieme a me crescerà e sarà capace di aiutarmi domani nel mio lavoro. Un semplice video-gioco, un valido home computer, un indispensabile personal? Lo decido io! E questo mi soddisfa.*

MPF II ha una struttura molto compatta e si avvale di soluzioni hardware originali ed espandibili. La più immediata è la tastiera esterna la cui connessione all'unità centrale è molto semplice.

Inoltre una serie di opzionali (disk drive, stampanti termiche, stampanti su carta normale, sintetizzatore vocale, monitor di formati diversi e con diversi tipi di fosfori, interfaccia seriale RS232C, joy-stick, generatore di suoni ed altro ancora) con i quali trasformi il tuo home computer in un personal professionale. Vuoi potenziare il tuo sistema informatico? Non devi ricominciare da capo. Sono tanti i connettori sui lati dell'MPF II che permettono di espanderlo fino a configurazioni estremamente potenti e già tutte attuabili.

Scegli tu!

Così hai la possibilità di divertirti, di studiare, di imparare il linguaggio Basic, sempre più importante. MPF II è accompagnato dai manuali d'uso e dal manuale di programmazione Basic tutti in lingua italiana. Un comodo ausilio di lavoro.

Il software è ampio e completo nelle tante cassette, nei dischi, nelle cartucce che vengono fornite insieme ad MPF II. È inoltre possibile accedere alla vasta bibliografia di programmi esistenti per la sua compatibilità di Basic...! MPF II, non scordiamolo, è dotato della tastiera incorporata e della scheda colore già installata. Tutto viene soddisfatto, i tuoi desideri, i tuoi giochi, le tue necessità, i tuoi lavori, la tua creatività. Pensa a qualcosa di grande per te, senza credere di sognare. MPF II è piccolo, leggero, ma ha grandi capacità di memoria e d'uso. Noi lo chiamiamo "l'investimento espandibile". E tu? Sceglilo e sarai al centro dell'attenzione di tutti.

Nella sua simpatica e morbida borsa da viaggio, insieme con tutti i componenti del sistema, viene sul lavoro, torna a casa, ti aiuta nello studio. Insomma MPF II è una scelta che ti dà soddisfazione, un sicuro investimento produttivo.



**CPU  
R 6502**

**ROM  
16K Bytes**

# 1. Il mio primo ed unico computer.



## Caratteristiche

L'unità centrale ha una tastiera alfanumerica di 49 tasti multifunzione con i quali c'è la possibilità di generare 153 codici ASCII.

È possibile il completo controllo del cursore tramite 4 appositi tasti. Lo schermo visualizza 24 righe per 40 colonne. Lavora con un set di caratteri ASCII maiuscolo e caratteri grafici speciali (50) raggiungibili dalla tastiera tramite il CTRL-B.

È disponibile una grafica contemporanea in 2 risoluzioni, high con 280x192 punti e low con 40x48 punti, a colori. È possibile miscelare testo e grafica.

Il microprocessore è il 6502. Sulla ROM è disponibile l'interprete Basic ed un monitor con disassemblatore per programmare anche in linguaggio macchina. L'altoparlante è presente.

L'unità centrale ha ben 64 K di memoria RAM dinamica e 16 K ROM. L'apposito slot porta all'esterno il BUS dati e indirizzi oltre ai segnali di controllo di tutto il computer. È possibile collegare interfacce e periferiche di tipo più svariato. L'unità centrale viene già fornita con un'interfaccia parallela per stampanti entro contenuta.



## MICRO-PROFESSOR MPF II

l'investimento espandibile

**RAM**  
64K Bytes

**Interprete Basic**  
più di 90  
istruzioni

Scrivici per ulteriori informazioni e per sapere dove puoi trovare MPF II vicino a casa tua.

PS 83

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Indirizzo \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**DIGITEK COMPUTER**

Ufficio Vendite  
Via Marmolada, 9/11 43068 SORBLOLO (Parma)  
Tel. 0521/69635 Telex 531083



# Tecniche di input e output per lo ZX Spectrum

## I segreti per sfruttare al meglio le caratteristiche meno note

di *Marcello Spero*

### Introduzione

Fin dalla presentazione dello ZX Spectrum sul mercato italiano era stato promesso l'arrivo "in tempi brevi" dei tanto sospirati Microdrive, ovvero le unità disco (anche se di vero e proprio disco non si può parlare) super economiche prodotte dalla stessa Sinclair (vedi *Personal Software n. 10-11* Ndr). A confortare le più ottimistiche speranze, una serie di comandi appositamente concepiti faceva bella mostra di sé sotto la prima fila di tasti.

Nell'attesa di una spiegazione "ufficiale" circa il loro uso, questi comandi possono essere vantaggiosamente utilizzati, almeno in parte, per risolvere alcuni problemi altrimenti insolubili se non ricorrendo al linguaggio macchina.

Infatti, se la maggior parte del software relativo all'impiego dei Microdrive sarà contenuto in una ROM all'interno dell'interfaccia relativa, nella ROM dello Spectrum sono già contenute le routine che riguardano la scelta del dispositivo cui inviare i dati prodotti da un programma o ricevere i dati necessari, mettendo in pratica il concetto, proprio di macchine di categoria più elevata, di file di ingresso e file di uscita.

All'interno di un programma BA-

SIC, o comunque usando i comandi di questo linguaggio, sappiamo dal manuale di poter usare:

PRINT per riferirci alla parte superiore dello schermo.

LPRINT per riferirci alla stampante ZX.

INPUT per introdurre dati dalla tastiera e visualizzarli nella parte bassa dello schermo.

### Linee di collegamento con le periferiche

La pratica, d'altra parte, ha dimostrato l'esistenza di un'altra forma per questi comandi:

PRINT # n;

LPRINT # n;

INPUT # n;

dove n è una qualsiasi espressione che dia un risultato numerico valido.

Normalmente dando ad n un valore diverso da 0, 1, 2 o 3 otterremo il messaggio di errore:

0 INVALID STREAM

Con il termine "stream" si vuole indicare il canale, o linea, di collegamento con le periferiche; n può quindi essere chiamato "numero di linea". (Da non confondere con il numero di linea dei programmi BASIC).

I quattro valori validi danno questi risultati:

0/1 Questi due valori sono equivalenti nel loro effetto: i dati contenuti nell'istruzione corrispondente, sia questa PRINT, LPRINT o INPUT, vengono visualizzati nella parte bassa dello schermo mentre, per la sola INPUT, vengono attesi dati in ingresso dalla tastiera.

2 Questo valore provoca la visualizzazione dei dati nella parte superiore dello schermo, senza

alcun riguardo al tipo di istruzione (PRINT, LPRINT, INPUT) usato.

Tentando invece l'immissione di dati, per esempio con:

INPUT # 2 ; a

otterremo il messaggio di errore:

J INVALID I/O DEVICE poiché la periferica associata alla linea 2 non è in grado di fornire dati in ingresso, come l'istruzione INPUT richiede. (Non notate anche voi qualcosa di strano nel modo in cui appare il messaggio di errore?).

3 Vale lo stesso discorso fatto per il 2, con la differenza che questa volta i dati sono inviati alla stampante invece che allo schermo.

### Utilizzo dei comandi

Un effetto immediato di questi comandi è la possibilità di offrire, nei programmi, la scelta fra video e stampante senza complicazioni:

100 INPUT "2 per l'output su video, 3 per l'output su stampante" ; n

...  
200 PRINT # n ; a ; b ; c ; "messaggio" ; d ; ...

...  
300 PRINT # n ; e ; f ; "messaggio" ; g ; ...

Un'applicazione più raffinata riguarda PRINT # 0 o PRINT # 1, mediante i quali possiamo visualizzare un messaggio alla base dello schermo, come se usassimo una INPUT, ma senza che questo sia seguito dal cursore in attesa di dati e quindi senza arrestare il corso del programma, proprio come una PRINT di tipo classico. Provate ad esempio:

PRINT # 0 ; "linea 24" : PAUSE 0 (l'istruzione PAUSE ha lo scopo di

## Tecniche di input e output per lo ZX Spectrum

evitare la conclusione del comando, poichè in questo caso il messaggio 0 OK si sovrapporrebbe a quanto abbiamo appena fatto stampare).

Nel corso di un programma un messaggio visualizzato nella parte inferiore dello schermo può essere facilmente cancellato con l'istruzione:

```
INPUT ""
```

che cancella la parte bassa dello schermo, non stampa niente e passa immediatamente il controllo all'istruzione successiva.

Provate:

```
10 FOR a = 1 TO 10
20 PRINT # 0 ; a
30 PAUSE 25
40 INPUT ""
50 NEXT a
```

Piuttosto utile può essere invece: 100 PRINT # 0; "premi un tasto per proseguire": PAUSE 0 : INPUT "" che evita di occupare con il messaggio la parte alta dello schermo.

Le righe effettivamente a disposizione nella parte bassa dello schermo sono due: quella normalmente utilizzata per i messaggi, cioè la linea 24, e la linea 23, normalmente vuota per separare le due parti dello schermo, ma accessibile mediante l'istruzione:

```
PRINT # 0 ; AT 0 , 0 ;
```

la linea 23 è infatti considerata la linea 0 della parte bassa, anche se le PRINT e le INPUT non indirizzate, si posizionano alla linea successiva. Eventuali PRINT ripetute non seguite da cancellazione, provocheranno, come nel caso di INPUT particolarmente voluminose, lo scrolling verso l'alto con la perdita delle prime linee dello schermo.

Nel modo descritto finora sono utilizzabili non solo le linee da 0 a 3 ma qualsiasi linea fino a 15, purchè ne informiate il sistema. Questo avviene con il comando:

```
OPEN n, c$
```

dove n è il numero della linea scelta, mentre c\$ è:

"P" per la stampante ZX,  
 "S" per la parte alta dello schermo,  
 "K" per la parte bassa dello schermo e l'immissione da tastiera.

Quindi includendo:

```
OPEN # 15, "P"
```

in un programma tutte le successive PRINT # 15 (o LPRINT # 15 o INPUT # 15) saranno dirette alla stampante.

Opposto è invece l'effetto del comando:

```
CLOSE # n
```

che cancella qualsiasi linea precedentemente aperta; unica eccezione sono le linee da 0 a 3, che non possono essere chiuse.

### La mappa di memoria dei canali

Ancora più interessante è vedere come questi comandi vengono interpretati dallo Spectrum.

Il manuale ci dà qualche indizio, nel capitolo "Variabili di sistema": qui infatti troviamo un blocco di 38 byte (23568 - 23605) chiamato "STRMS" che contiene gli "indirizzi dei canali collegati alle linee (stream)"; agli indirizzi 23631/2 troviamo invece la variabile "CHANS" che contiene l'"indirizzo dei dati riguardanti i canali", seguita agli indirizzi 23633/4 da CURCHL", l'"indirizzo delle informazioni usate al momento per le operazioni di ingresso e uscita". Infine la mappa della memoria del capitolo 24 rivela un'area definita "informazioni sui canali", il cui inizio è contenuto nella variabile CHANS, mentre il termine viene segnato da un byte contenente 80 Hex (esadecimale).

Alcune indagini circa il contenuto di queste locazioni di memoria nonché sul modo di funzionamento delle routine ROM interessate, hanno

portato a questo quadro della situazione: i 38 byte che compongono STRMS sono suddivisi in 19 gruppi di due byte ciascuno che all'atto dell'accensione o dopo il comando NEW assumono la configurazione illustrata in figura 1. I primi tre gruppi vengono usati dalla ROM per scopi particolari, mentre i successivi 16 corrispondono alle linee da 0 a 15. I comandi OPEN # e CLOSE # modificano questi valori; OPEN # 4, "S" ad esempio, modifica il valore del gruppo 23582/3, portandolo a 6.

I valori contenuti nei vari gruppi sono in realtà dei puntatori: sommando il valore contenuto nella variabile CHANS al valore del gruppo corrispondente alla linea prescelta e sottraendo 1 otteniamo l'indirizzo di ingresso di un particolare blocco all'interno dell'area "informazioni sui canali" della RAM. In quest'area ciascun blocco è composto da cinque byte e consiste di due gruppi di due byte contenenti ciascuno un indirizzo, seguiti da un byte contenente il codice di uno dei caratteri K, S, R o P (K, S e P sono, guarda caso, le lettere consentite nei comandi OPEN # e CLOSE #).

Allo stato attuale delle cose, cioè senza il complesso RS232/rete/Microdrive, l'area di informazione sui canali è lunga 21 byte e contiene quattro blocchi, come si vede in figura 2. Il primo indirizzo di ciascun blocco è quello di inizio di una routine ROM che viene utilizzata per l'emissione dei caratteri; il secondo indirizzo è, a seconda dei casi, quello della routine ROM per l'immissione dei caratteri da tastiera oppure della routine che produce il messaggio di errore "J INVALID I/O DEVICE".

Confrontando le due figure ci si rende conto che la prima coppia di

## Tecniche di input e output per lo ZX Spectrum

indirizzi dell'area di informazione sui canali viene usata dalle linee 0 e 1, la seconda dalla linea 2 e la quarta dalla linea 3. In ciascun caso lo Spectrum trova l'indirizzo della appropriata routine di I/O alla posizione corrispondente dell'area di informazione e quindi la utilizza (mediante una istruzione CALL in linguaggio macchina), ponendo nel registro A dello Z80 il carattere da emettere. Tutto questo per dire che se volessimo, ad esempio, utilizzare una stampante non ZX, potremmo scrivere una routine in linguaggio macchina per pilotarla e quindi sostituire i primi due byte del quarto blocco nell'area di informazione con l'indirizzo di inizio di questa nuova routine. In questo modo qualsiasi istruzione LPRINT, LLIST o PRINT # 3 userà automaticamente la nostra routine. A quanti volessero realmente cimentarsi in questa non facile operazione ricordo che la routine dovrà essere in grado di riconoscere ed utilizzare in modo appropriato sia i caratteri di controllo della stampa (dal 6 al 23) che i codici corrispondenti alle varie parole del BASIC (dal 165 al 255), nonché sbrigarcela con i vari caratteri grafici, definibili e non.

Osservando la figura 2 notiamo che la routine di emissione cui viene fatto riferimento è sempre la stessa (parte dall'indirizzo 09F4), sia che si debba stampare nella parte alta che in quella bassa dello schermo, ed è ancora la stessa per la stampante. Questo può avvenire perché la routine sceglie il destinatario dei caratteri in uscita in base allo stato del bit uno della variabile di sistema FLAGS (23611): 0 se l'output è diretto allo schermo, 1 se alla stampante; il bit zero della variabile TVFLAG (23612), inoltre, è 0 per la parte alta dello schermo e 1 per la parte bassa.

### La routine di emissione dei caratteri

Per concludere, una parola sul

LOCAZIONE	CONTENUTO	LINEA (STREAM)
23568	1	—
23570	6	—
23572	11	—
23574	1	0
23576	1	1
23578	6	2
23580	16	3

Le locazioni da 23581 a 23605 contengono zero

Figura 1. Contenuto di STRMS all'accensione o dopo l'esecuzione del comando NEW.

CODICI HEX E SIGNIFICATO DEI 21 BYTE DELL'AREA	
F4	} Indirizzo della routine di PRINT.
09	
AB	} Indirizzo della routine di ingresso dei caratteri da tastiera.
10	
4B	Codice della lettera K.
F4	} Indirizzo della routine di PRINT.
09	
C4	} Indirizzo della routine di errore J INVALID I/O DEVICE.
15	
53	Codice della lettera S.
81	} Indirizzo della routine per inserire caratteri nella RAM.
0F	
C4	} Indirizzo della routine di errore J INVALID I/O DEVICE.
15	
F4	Codice della lettera R.
F4	} Indirizzo della routine di PRINT.
09	
C4	} Indirizzo della routine di errore J INVALID I/O DEVICE.
15	
50	Codice della lettera P.
80	Fine dell'area.

Figura 2. Mappa dell'area contenente le informazioni sui canali.

funzionamento della routine di emissione dei caratteri, molto usata da chi programma in linguaggio macchina, che viene utilizzata mediante l'istruzione RST 10 dopo aver posto nel registro A il codice del carattere da stampare.

Questa routine, in effetti, prende il contenuto di CURCHL, cioè l'indirizzo di un blocco all'interno dell'area di informazione sui canali, e quindi salta all'indirizzo indicato dal blocco in questione; a questo

punto FLAGS e TVFLAG determinano la destinazione dell'output. Questi flag sono peraltro facilmente modificabili per mezzo della routine ROM che inizia all'indirizzo 1601 Hex, avendo cura, prima di chiamare la routine, di porre nel registro A uno dei seguenti valori:

- 0 per l'uscita nella parte bassa dello schermo,
- 2 per l'uscita nella parte alta dello schermo,
- 3 per l'uscita su stampante. ■

espande all'infinito la tua esperienza

HES

REBIT  
COMPUTER  
A DIVISION OF G.B.C.

Synthesound

Gridrunner  
By Jeff Meier

Cartridge for VIC 20

IRTH

language that is  
BASIC, you can  
write and save  
your programs  
and print them  
on a printer.



Cartridge for VIC 20



HES WRITER  
Cartridge for VIC 20

HES WRITER  
HES WRITER  
HES WRITER

Shamus  
Cartridge for VIC 20



Protector  
Cartridge for VIC 20

Cartridge for VIC 20

Shamus  
Cartridge for VIC 20

HES MON  
Cartridge for VIC 20

HES MON is a 3000 machine language  
interpreter for the VIC 20. It allows you to  
write programs in BASIC and save them  
on disk. It also allows you to load and  
run programs written in other  
languages. (See text on the back of the  
box for more details.)  
RAM expansion is optional.  
Complete instruction manual is included.

HES MON

HES MON

Cartridge for Commodore 64

software a misura d'uomo

# Programmer's tool kit per Commodore 64

— prima parte —

## Un potente strumento per sviluppare il vostro software

di *Alessandro Guida*

**P**ubblichiamo in questo numero la prima parte di un impegnativo lavoro sviluppato dal nostro collaboratore Alessandro Guida. Speriamo così di acccontentare i numerosi lettori che ci hanno chiesto software per il C64, fornendoli di una utility di indubbio interesse ed elevato valore commerciale.

### Cos'è un tool kit

La traduzione più o meno letterale sarebbe "attrezzi del mestiere". Infatti si tratta proprio di questo: una raccolta di utilities che rendano meno gravoso il compito di chi deve programmare.

Grazie al programma proposto avremo a disposizione nuovi comandi che permettono una più facile stesura e correzione dei programmi in BASIC.

### Il programmer's tool kit

Trattandosi di un programma scritto interamente in linguaggio macchina potrebbe risultare ostico a chi non ha avuto occasione di lavorare in linguaggio oggetto.

D'altronde non è necessario capire come è scritto per poterlo utilizzare traendone i migliori risultati.

Perciò parleremo principalmente

dei comandi e relative sintassi, e nel prossimo numero, per i più interessati, daremo un'occhiata alle routine in linguaggio macchina.

Come avrete visto il nostro tool kit è messo sotto forma di frasi DATA in un programma BASIC. Digitatelo nel vostro 64 e, dopo averlo salvato su nastro o dischetto, date il RUN. Il programma caricherà il sottoprogramma in linguaggio macchina, aggiornerà i puntatori necessari e al termine si autocancellerà.

Infatti a questo punto avremo il P.T.K. già in funzione e non sarà più necessario conservare il programma sorgente in BASIC.

Va aggiunto che l'installazione del tool kit toglie solamente 1,7 Kbyte ai 38 Kbyte disponibili all'acensione.

### I nuovi comandi

Nella presentazione della sintassi dei comandi le parti in maiuscolo vanno scritte nella stessa maniera, quelle tra parentesi sono opzionali e quelle in minuscolo vanno sostituite con i vostri valori.

I nuovi comandi di cui ora vedremo la sintassi vanno dati in modo diretto e non inseriti nei programmi. Inoltre non va dato più di uno di questi comandi per linea.

**HELP:** nel caso si sia verificato un errore nel vostro programma dando questo comando vedrete sullo schermo la linea dell'errore e il punto in cui questo si è verificato, segnalato da un punto interrogativo.

Sintassi : HELP.  
Es.:

SNTAX ERROR IN 50  
READY.

HELP

50 FORI = 0T050 : PLINT ? I :  
NEXT  
READY.

**AUTO:** permette la numerazione automatica delle linee BASIC in fase di scrittura del programma.

Sintassi : AUTO (inizio,) (incremento).

Es.:

AUTO           numera il programma partendo da 10 con un incremento di 10.

AUTO100,       parte da 100 e incrementa di 10.

AUTO, 20       parte da 10 e incrementa di 20.

AUTO100, 20   parte da 100 e incrementa di 20.

AUTO0, 0       disabilita la numerazione automat.

**FIND:** cerca una stringa, un testo, una variabile o degli statement all'interno del programma, listando tutte le linee che contengono il testo cercato.

Sintassi : FIND, (") testo (").

Es.:

FIND, A\$       cerca tutte le variabili A\$ nel programma.

FIND,  
"PIPP0"       cerca la stringa PIPPO.

FIND,  
GOTO100       cerca lo statement GOTO100.

**DUMP:** lista tutte le variabili usate nel programma con il loro ultimo valore.

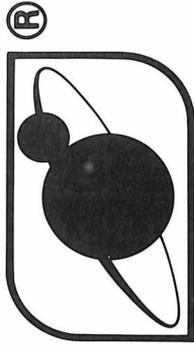
**NOTA:** perchè le variabili vengano stampate il programma che le contiene deve essere stato eseguito.

Sintassi : DUMP.





# BREMI®



**qualità e prezzo  
nei monitor professionali**

- professional computer
- business
- personal

Stilo R NENICI - PARMA



#### **MONITOR DA 14 POLLICI COLORI PER DATA GRAPHIC DISPLAY**

Caratteristiche: dell'ingresso per segnale composito video; impedenza d'ingresso: 75 Ω; banda passante: 20 MHz.

Caratteristiche dell'ingresso per segnali TTL separati: banda passante: 5 MHz.

Alimentazione: 220/240 V AC

Frequenze di scansione: orizzontale: 15.700 ± 500 Hz; verticale: compresa tra 47 e 62 Hz.

Angolo di deflessione: 30°.

Distorsione: orizzontale: 3%.

Durata dello spegnimento orizzontale: 12 µsec.

Durata dello spegnimento verticale: 1,2 msec.

Dimensioni: 440 × 470 × 440 mm.

#### **MONITOR DA 12 POLLICI PER DATA GRAPHIC DISPLAY**

Cinescopio a fosfori verdi, bianchi o arancio a persistenza medio-breve. Due modelli disponibili:

1) monitor con ingresso per segnale composito video; banda passante 24 MHz, impedenza d'ingresso 75 Ω,

alimentazione 220/240 V AC 20 W.

2) monitor con ingresso per segnali TTL separati (positivi e negativi); banda passante 30 MHz.

Frequenza di scansione: orizzontale: 15.700 ± 500 Hz; verticale: compresa tra 47 e 62 Hz.

Angolo di deflessione: 30°.

Distorsione: orizzontale: 3%.

Tolleranza di deflessione: 3%.

Numero di caratteri: 2000 (80 × 25 linee).

Arco del display: 250 × 185 mm.

Modelli disponibili con o senza contenitore.

Dimensioni con contenitore: 400 × 390 × 380 mm. Dimensioni senza contenitore: 290 × 240 × 300 mm.

BREMI division data

43100 PARMA (ITALIA) - Via Benedetta 165/A Tel. 0521/72209-75680-771533-771284 ● Tx 531304 Bremi-1



**E ADESSO CHE HO COMPRATO  
IL MIO PRIMO REGISTRO IVA,  
CHI MI AIUTERA' A FARE IN MODO  
CHE NON SIA ANCHE L'ULTIMO?**



# IL PERSONAL COMPUTER IBM IL TUO PICCOLO GRANDE AMICO.

Bene, la tua nuova attività è nata. E subito incominciano i problemi con la fatture, la contabilità, i registri. Insomma, un mucchio di complicazioni che rischiano di distoglierti proprio dalle fasi più importanti dell'avvio del nuovo lavoro.

Ma oggi, per fortuna, c'è un amico pronto a darti una mano: il Personal Computer IBM. Così piccolo da stare sulla tua scrivania, tanto grande da aiutarti a risolvere tutti i problemi di IVA. E non solo quelli.

Perché il Personal Computer IBM può fare di tutto: riceve dati, calcola, fa statistiche, registra, controlla, archivia e stampa. E non è necessario essere un addetto ai lavori per imparare ad usarlo. Vedrai, in poche ore diventerete ottimi amici. Perché ragioni come te.

Vuoi metterlo alla prova? Vai da un concessionario IBM. Scegli quello che ti è più comodo nell'elenco della pagina che segue.



IBM Italia  
Distribuzione Prodotti srl



Il Personal Computer IBM contiene un microprocessore a 16 bit e una memoria di utilizzo che raggiunge i 640 Kbyte, e può essere dotato di un video a colori e di un processore matematico. E, grazie ai dischi fissi, la capacità massima di memoria del sistema è di 21 Mbyte in linea. Inoltre, puoi facilmente collegarti con un altro Personal Computer IBM, con elaboratori più potenti e con la rete dei Centri Servizi Elaborazione Dati della IBM.

**Sistemi operativi:** DOS 1 - DOS 2 - UCSD - CP/M-86. **Supporti per le comunicazioni:** Asincrono - SDLC - BSC - Emulazione: 3101-3270. **Linguaggi:** tutti i principali e in più l'APL. **Programmi applicativi per:** aziende e servizi - produttività individuale - ufficio moderno - calcolo tecnico e scientifico - applicazioni professionali - didattica.

# Asteroidi

## per ZX Spectrum

### Un classico gioco spaziale ora anche per Sinclair

di Marcello Spero

**Q**uesto gioco ricalca un noto videogame che molti di voi senz'altro conosceranno: al comando di un veicolo spaziale, dovette resistere il più a lungo possibile muovendovi in un denso ammasso di asteroidi.

A disposizione avete, oltre alla vostra abilità di manovra, uno schermo di energia che potrà essere usato per un numero limitatissimo di volte, e quindi solo per i casi più disperati. Dopo un certo numero di collisioni il veicolo si disintegrerà. Le istruzioni complete appaiono prima dell'inizio del gioco; ricordo soltanto che avete la possibilità di ottenere un "bonus" ogni duecento punti, aumentando considerevolmente le vostre possibilità di sopravvivenza.

Al termine del gioco vengono visualizzati tre punteggi: quello della partita appena conclusa, il migliore dal momento in cui il gioco è stato caricato ed il record assoluto. Quest'ultimo viene registrato, insieme alle iniziali del detentore, in modo da essere riprodotto ad ogni successivo caricamento del gioco, finché non sarà superato. In questo caso vi sarà offerta l'opportunità di registrare il nuovo record con relative iniziali.

Qualche nota al programma: per il movimento dello schermo è stata utilizzata la routine di scroll verso l'alto propria della macchina, mentre per quello della nave la linea 50 attua un procedimento logico abbastanza noto, che ha il vantaggio di

impegnare sempre lo stesso tempo, indipendentemente dal fatto che avvenga o meno il movimento.

Il riconoscimento delle collisioni si ottiene con la funzione ATTR in linea 60 e 65, in cui si distingue fra un urto laterale o centrale, ai fini del punteggio.

Completano il gioco alcuni effetti sonori, progettati in modo da non rallentare l'azione.

I caratteri alfabetici maiuscoli contenuti fra virgolette sulle linee 55, 110, 400 e 420 sono caratteri grafici e vanno quindi ottenuti con il cursore "G"; servono a rappresentare l'astronave, il campo di forza e l'esplosione. Gli asteroidi vengono invece visualizzati dalla linea 120 in modo casuale, sia per carattere che per posizione. Il caricamento dei dati relativi alla grafica avviene mediante la subroutine che inizia alla linea 9000, e viene richiamata subito dopo la visualizzazione delle istruzioni.

Per salvare il programma è consigliabile usare il comando diretto:

SAVE "asteroidi" LINE 1

che ne provoca la partenza automatica non appena terminato il caricamento.

Un'ultima nota importante riguarda il primo utilizzo del programma, quando cioè non è stato ancora registrato alcun record. Non appena

inizia l'esecuzione il computer si pone in attesa di dati dal registratore (situazione riconoscibile dal combinazione di colore del "border"), poiché è prevista la presenza, in coda al programma, di un "number array" contenente i dati relativi al record (valore e iniziali). All'atto del primo utilizzo, però, questi dati non esistono ancora, e l'esecuzione non procede oltre. Premete allora il BREAK (è sufficiente il solo tasto dello spazio) e quindi fate ripartire il programma con un comando:

GO TO 3

che scavalca lo stadio di caricamento dati; in questo modo il valore del record viene posto a zero. Al termine della partita vi verrà detto che avete battute il record (ovviamente!) e potrete quindi registrarlo, possibilmente subito dopo il programma. Da questo momento l'esecuzione potrà avvenire in modo normale, completamente automatico; gli eventuali aggiornamenti del record andranno registrati in modo da sovrapporsi ai dati precedenti, che vanno cancellati.

Non è necessario far notare che sono possibili infiniti sviluppi e ampliamenti alla struttura del gioco, che qui sono stati evitati per contenere l'estensione del listato entro limiti ragionevoli: a voi l'iniziativa, e buon divertimento!

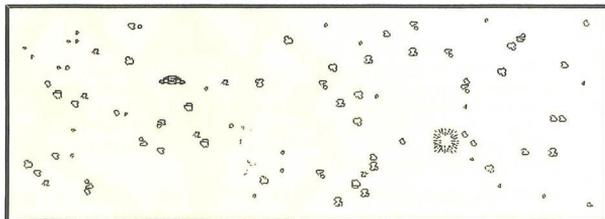


Figura 1. Due esempi di uscita su stampante.

# ECC CHI TI AIUTERA' AD ANDARE D'AMORE E D'ACCORDO CON IL TUO NUOVO AMICO



Il tuo concessionario IBM. Ti aiuterà a ottenere il massimo dal tuo Personal Computer IBM. Ti garantirà un'assistenza puntuale e un servizio all'altezza del nome IBM, che in tutto il mondo significa efficienza e affidabilità. Per una lunga e proficua amicizia fra te e il tuo Personal Computer IBM. Per acquisti superiori alle 20 unità puoi anche rivolgerti alle filiali IBM. E per ulteriori informazioni sui eventuali punti di vendita che non compaiono sull'elenco, telefona a: 02/21752360 oppure 06/54864962.

## ABRUZZO/MOLISE

Pescara - ITALDATA SRL - Via Tiburtina, 75 - Tel. 052426460  
 - Tel. 05240494800  
 Campobasso - PUBLISISTEMI SRL - Via S. Antonio Abate, 236 - Tel. 0874.98144

## BASILICATA

Potenza - I.P.E.S. SPA - Via Sanremo, 79 - Tel. 0971.43293

## CALABRIA

Cosenza - CALIÒ SRL - Via N. Serra, 90 - Tel. 0984.32807

## COMPANIA

Cava dei Tirreni - METELLIANA SPA - Via Mandoli, 16  
 Tel. 089.46587  
 Napoli - POINT SYSTEMI SRL - Via A. De Gasperi, 45  
 Tel. 081.312312  
 Salerno - OMNIA SRL - C.so Garibaldi, 47 - Tel. 089.2200636  
 S. Maria Capuverture - GENERALI SYSTEMS S.R.L.  
 Via Italia d'Italia, 21/23 - Tel. 0823.811100

## EMILIA

Bologna  
 ARAI SAS - Via Bernini, 1 - Tel. 051.932374  
 CMB INFORMATICA SCRL - Via Arcoveggio, 74/10 - Tel. 051.323594  
 LICOM SYSTEMS SRL - Via Farini, 33/A - Tel. 051.231569  
 SYSDATA ITALIA SPA - Via Massimo d'Azeglio, 58 - Tel. 051.330021  
 Corpi  
 DATA SRL - Via B. Peruzzi, 12 - Tel. 059.688900  
 UNIDATAN SRL - Via Biadoli, 6 - Tel. 059.698355  
 Ferrar - MARKITALIA COMPUTERS SRL - Via Bologna, 84 - Tel. 0532.33867

## Fiori

C.E.D.A.F. COOP. ELAB. DATI - Via Zenchini, 57 - Tel. 0543.652402  
 I.C.O.T. IMPIANTI SRL - Via Codazzi, 10 - Tel. 0543.723014  
 ITALIA - PALAZZO DONATO - Via Emilia, 23/38 - Tel. 0542.29195  
 Piacenza - RCM COMPUTER SAS - C.so Vittorio Emanuele II, 96 - Tel. 0523.37358

## Reggio Emilia

A.P.E.D. ELABORAZIONE DATI - Via Filippo Re, 17 - Tel. 0522.365721

## MEMAR ELECTRONIC SRL - V.le Melato, 13 - Tel. 0522.94230

Rimini - HARD & SOFT SYSTEMS SRL - Via Valturio, 43  
 Tel. 0541.773443

## LAZIO

Frosinone - SAU ELETTRONICA SRL - Via Vado del Tufo, 85 - Tel. 0775.53093

## Roma

CERVED SPA - Via Appia Nuova, 696 - Tel. 06.7940241  
 DATAOFFICE SPA - Via Sicilia, 205 - Tel. 06.4754568  
 ELEDRA SS SPA - Via G. Valmarana, 63 - Tel. 06.8127324  
 GEDIN SRL - L.go D. De Dominicis, 7 - Tel. 06.432183  
 I.S.E.D. SPA - Via Tiburtina, 1296 - Tel. 06.4256851  
 I.R.T. SISTEMI PER L'INFORMATICA SPA - P.zza SS Apostoli, 66 - Tel. 06.6793477  
 JACOROSI SPA - Via V. Brancati, 84 - Tel. 06.50091  
 MEMORY COMPUTER SRL - Via Aureliana, 39 - P.zza S. 06.804592/4755756  
 MICROCOMPUTER - Via M. Gelsomini, 28/30 - Tel. 06.577484/324  
 NICA DIFF INF SRL - V.le Paroli, 40 - Tel. 06.872603  
 SPESER SRL - V.le Tito Livio, 12 - Tel. 06.3465356  
 SUDADEL SRL - P.zza S. Anastasia, 3 - Tel. 06.6786663  
 Viterbo - ITALBYTE SRL - V.le Trento - P.l. Garbini - Tel. 0761.221333

## LIGURIA

Genova  
 DIFEL SRL - Via XX Settembre, 31/4 - Tel. 010.586288  
 ELABORATION PROCESS SRL - Via Fieschi, 3/6 - Tel. 010.565704/58986  
 I.P.E.S. SRL - Via SS. Giacomo e Filippo, 13R - Tel. 010.87344/4  
 Sanremo - DIFFEL - C/O D.RCS SRL - Via Heliosino, 8 - Tel. 0184.72435

## LOMBARDIA

Novara - NUOVA INFORMATICA SAS - Via Provinciale, 86 Comendano - Tel. 035.751784

Assago - TRANSDATA SRL - Mi Fiori Pl. E3 Str. 1 - Tel. 02.8246460  
 Bergamo - SELTERING SPA - Via Verdi, 31 - Tel. 035.248256

## Brescia

FIN-ECO SERVICE SRL - Via Pastrengo, 5 - Tel. 030.590955  
 MICROSELT SRL - Via Cipro, 33 - Tel. 030.224246  
 SELTERING SPA - Via Cipro, 33 - Tel. 030.2242391  
 Como - BRUNO SRL - Via Rubini, 5 - Tel. 031.260538  
 Lecco - ZECCA UFFICIO SPA - Viale Dante, 14 - Tel. 031.4372291

## Lodi - ZUCCHETTI SPA - C.so Mazzini, 39 - Tel. 0371.54827

Mantova - REPLICA COMPUTER SRL - Via G. Arrivabene, 22  
 Tel. 0376.98821/2

## Milano

B.M.O. SYSTEM SAS - Via Desenzano, 7 - Tel. 02.4080275  
 B.M. SAS - V.le Tunisia, 50 - Tel. 02.6599870  
 C.S.A. COMM. SRL - Via Farini, 82 - Tel. 02.6888433  
 DATA OPTIMIZATION SPA - Via Masaccio, 12 - Tel. 02.4987876

## ECN ITALIA SRL - C.so Monforte, 15 - Tel. 02.780213

EDLEKTRON SRL - C.so Scarpone, 39 - Tel. 02.2439603  
 ELEDRA SS SPA - Viale Elvezia, 14 - Tel. 02.349751  
 GENERAL ELECTRIC INFORMATION SERVICES SPA - Piazza Genova, 29 - Tel. 02.2870181  
 HOMIC PERSONAL COMPUTER SRL - Piazza De Angelis, 3 - Tel. 02.4986201

## HUGNOT LUIGI LUCIANO - Via De Togni, 10 - Tel. 02.873190

IL NUOVO UFFICIO SISTEMI SNC - Via Priv. del Don, 2 - Tel. 02.839780

## MICROTECH SRL - Via F.lli Bronzetti, 20 - Tel. 02.736690

SIRIO SHOP SRL - Viale Cortosa, 148 - Tel. 301061  
 SOFTEC SRL - Viale Manno, 10 - Tel. 02.7491196

## S.D.I. STUDIO DI INFORMATICA SPA

Via G. Winckelmann, 11 - Tel. 02.4223305  
 Monza - E.D.CONSULI SRL - Via Rosmini, 3 - Tel. 039.389850

## Napoli

S. Antonio Mantov. - ANTEK COMPUTER SAS - Via Manzoni, 49 - Tel. 081.6398759  
 Sondrio - G.P.D. SRL OFF. AUTOM. - V.le N. Saura, 28 - Tel. 0342.218651

## Napoli

ELMEC SPA - Via Sebenico, 12 - Tel. 0332.264135  
 VEGA SPA - Via Silvestro Sanvito, 103 - Tel. 0332.229374  
 Vigevano - LOGICA INFORMATICA SRL - Via Montegrappa, 31 - Tel. 0381.81838

## Vimerate - DATA PROGRES SRL - Via V. Emanuele, 44/4 - Tel. 039.667423

## Vimodrone - OMEGA DATA SRL - Strada Padana Spv, 317 - Tel. 02.2504121

## MARCHE

Jesi - SYST. HOUSE A.P.R.A. SRL - V.le Cavallotti, 9 - Tel. 073.85743  
 Moie - S.E.D.A. SPA - P.zza S. Maria - Tel. 0731.70345/7603  
 Pesaro - COMPUTER & OFFICE SRL - Via Mazzini, 70 - Tel. 073.614190

## PIEMONTE

Alessandria - INFORMATICA SERVICE SRL - Via Isorno, 63 - Tel. 031.445817  
 Asti - IASTIA DATATI SNC - Via Silvio Morando, 6/A - Tel. 0141.263556

## Biella

TEOREMA SRL - Via Losana, 9 - Tel. 015.24915  
 V.I. COMPUTERS SRL - Via Repubblica, 39 - Tel. 015.274106

## Bergamo - I.D.S. INF. DATA SYST. SRL - V.le Varolio, 157 - Tel. 0363.25327

Cuneo - SISTEMI SRL - Via Giolitti, 26 - Tel. 0171.55475  
 Genova - EURO SISTEMI SPA - Bova SS, 30/28 - Tel. 0172.88176

## DIVERSIFICATA VENCO SRL - C.so Matteotti, 32A - Tel. 011.545525

PROGRAMMA SPA - C.so Svizzera, 185 - Tel. 011.746421  
 SISTEMI SPA - C.so Peschiera, 249 - Tel. 011.338676

SOFTEC SRL - C.so San Maurizio, 79 - Tel. 011.8396444  
 Vercelli - ANALOG SNC - Via Dionisotti, 18 - Tel. 0161.61105

## PUGLIA

Bari - PASEL SRL - Via Calefati, 124/136 - Tel. 080.481488  
 Foggia - MASELLI PER L'UFFICIO SPA - Via L. Zuppetta, 355A  
 Tel. 0881.76014  
 Lecce - I.P.E.S. SPA - Via Oberdan, 29 - Tel. 0832.33904  
 Maglie - S.V.I.C. SRL - Via V. Emanuele, 121 - Tel. 0836.216204

## SARDEGNA

Cagliari - G.D.S. SAS - Via Sonnino, 108 - Tel. 070.650756

## SICILIA

Catania  
 ASIA COMPUTER SRL - Via S. Euplio, 13 - Tel. 095.32844  
 COMPUTER SYSTEMS SRL - Via Ruggero di Lauria, 87 - Tel. 095.493777  
 Messina - SICIL FORNITURE SPA - Via Don Basso, 75 - Tel. 090.2923887  
 Palermo  
 SERCOM ITALIA SRL - Via Scitù, 180 - Tel. 091.261041  
 SIRI SRL - Via Serradellino, 145 - Tel. 091.577344  
 TESI SRL - Via R. Napolitano, 23 - Tel. 091.260549  
 Trapani - TESI SRL - Via Palmiero Abate, 17 - Tel. 0923.20226\*

## TOSCANA

Empoli - SESA DISTRIBUZIONE SRL - Via N. Febbraio, 24/B - Tel. 0571.72148

## Firenze

DATA COOP. SRL - Via di Novoli, 27/15 - Tel. 055.478966  
 SESA DISTRIBUZIONE SRL - Lungarno Ferrucci, 19R - Tel. 055.681862  
 SFC COM. SAS - Viale Garibaldi, 298 - Tel. 0574.580222  
 Siena - SILOG SRL - Via Sicilia, 5 - Belverde - Tel. 0577.94065  
 Viareggio - DELPHI SRL - Via Aurelia Sud, 39 - Tel. 0544.989161

## TRIVENETO

Bassano D/Grappa - C.P.E. - Piazzetta Poste, 9 - Tel. 0424.20395

## Belluno

DE PRA SRL - Via I. Caffi, 18 - Tel. 0437.23243/28089

## BOLOGNA

BOLOGNA - BOPAM SAS - Via C. Battisti, 32 - Tel. 0471.30131  
 Castelfranco Ven. - EDS SRL - Via S. Pio X, 154 - Tel. 0423.490178

## Padova

CERVED ENGINEERING SPA - C.so Stati Uniti, 14 - Tel. 049.760733  
 S.I.C. ITALIA SRL - Via Fiammbra, 8 - Tel. 049.455556  
 SYSTEM ROS SAS - P.zza De Gasperi, 14 - Tel. 049.38412

## SOLINGA SPA - Via Marsala, 29 - Tel. 049.655385/657386

S. Donà di Piave - COMPUTIME SRL - Piazza Rizzo, 63 - Tel. 0421.2546

## Trento

SEDA SAS - Via Sighere, 71 - Tel. 0461.984564  
 SGE SNC - COMPUTER SHOP - Via Prato, 22 - Tel. 0461.25154

## Trivigno - INFORMATICA TRE SRL - Viale della Repubblica, 18 - Tel. 0422.65993

## Trieste - DITTA MURRI - Via A. Diaz, 24/A - Tel. 040.733253

## Udine

D.E.U. SRL - Via Di Prampero, 3/7 - Tel. 0432.204032  
 D.E.U. SRL - Via Tavagnano, 89 - Tel. 0432.480806

## Venezia

PRAGMA SOFTWARE SRL - Via Carmelitani Sacri, 20 - Tel. 041.24629  
 SEVER DI G. SERENI - Via Locatelli, 10 - Tel. 041.33331  
 VICENZA - ALFA DATA SRL - Via Milano, 110 - Tel. 0444.81865

## Verona

Perugia - PUCCIUFFINO SNC - Via XX Settembre, 148/C  
 Tel. 073.72992  
 Pinerolo - D.F.S. SRL - Via Piacinotti, 6 - Tel. 0744.82847

## VALDAOSTA

Aosta  
 INFORMATIQUE SAS - Av. Du Cons. De Commis, 16 - Tel. 0165.2242

Per maggiori informazioni, compila e spedisce questo tagliando  
 al tuo concessionario di zona.

Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Via \_\_\_\_\_

N° \_\_\_\_\_

Cap \_\_\_\_\_ Città \_\_\_\_\_

IBM Italia  
 Distribuzione Prodotti srl

Personal Software 12/13



# in edicola

- Prezzi
- Caratteristiche
- Descrizioni
- Prove



- Tutti i videogames
- Tutte le console
- Tutti i giochini tascabili
- Tutti gli accessori
- Tutti i giocomputer
- Tutti a colori

**Tutto... tutto...  
ma proprio tutto.**



## Asteroidi per ZX Spectrum

### Listato 1. Il programma Asteroidi.

```

1 REM
2 *****
3 A S T E R O I D I
4 *****
5
6 BORDER 1: PAPER 1: CLS : IN
7 DATA LET B=0: DIM e(13): LOAD "
8 CLS : FOR I=2 TO 7: PRINT A
9 T B: GOTO 10
10 PRINT
11 *****
12 *****
13 *****
14 *****
15 *****
16 *****
17 *****
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
101 *****
102 *****
103 *****
104 *****
105 *****
106 *****
107 *****
108 *****
109 *****
110 *****
111 *****
112 *****
113 *****
114 *****
115 *****
116 *****
117 *****
118 *****
119 *****
120 *****
121 *****
122 *****
123 *****
124 *****
125 *****
126 *****
127 *****
128 *****
129 *****
130 *****
131 *****
132 *****
133 *****
134 *****
135 *****
136 *****
137 *****
138 *****
139 *****
140 *****
141 *****
142 *****
143 *****
144 *****
145 *****
146 *****
147 *****
148 *****
149 *****
150 *****
151 *****
152 *****
153 *****
154 *****
155 *****
156 *****
157 *****
158 *****
159 *****
160 *****
161 *****
162 *****
163 *****
164 *****
165 *****
166 *****
167 *****
168 *****
169 *****
170 *****
171 *****
172 *****
173 *****
174 *****
175 *****
176 *****
177 *****
178 *****
179 *****
180 *****
181 *****
182 *****
183 *****
184 *****
185 *****
186 *****
187 *****
188 *****
189 *****
190 *****
191 *****
192 *****
193 *****
194 *****
195 *****
196 *****
197 *****
198 *****
199 *****
200 *****
201 *****
202 *****
203 *****
204 *****
205 *****
206 *****
207 *****
208 *****
209 *****
210 *****
211 *****
212 *****
213 *****
214 *****
215 *****
216 *****
217 *****
218 *****
219 *****
220 *****
221 *****
222 *****
223 *****
224 *****
225 *****
226 *****
227 *****
228 *****
229 *****
230 *****
231 *****
232 *****
233 *****
234 *****
235 *****
236 *****
237 *****
238 *****
239 *****
240 *****
241 *****
242 *****
243 *****
244 *****
245 *****
246 *****
247 *****
248 *****
249 *****
250 *****
251 *****
252 *****
253 *****
254 *****
255 *****
256 *****
257 *****
258 *****
259 *****
260 *****
261 *****
262 *****
263 *****
264 *****
265 *****
266 *****
267 *****
268 *****
269 *****
270 *****
271 *****
272 *****
273 *****
274 *****
275 *****
276 *****
277 *****
278 *****
279 *****
280 *****
281 *****
282 *****
283 *****
284 *****
285 *****
286 *****
287 *****
288 *****
289 *****
290 *****
291 *****
292 *****
293 *****
294 *****
295 *****
296 *****
297 *****
298 *****
299 *****
300 *****
301 *****
302 *****
303 *****
304 *****
305 *****
306 *****
307 *****
308 *****
309 *****
310 *****
311 *****
312 *****
313 *****
314 *****
315 *****
316 *****
317 *****
318 *****
319 *****
320 *****
321 *****
322 *****
323 *****
324 *****
325 *****
326 *****
327 *****
328 *****
329 *****
330 *****
331 *****
332 *****
333 *****
334 *****
335 *****
336 *****
337 *****
338 *****
339 *****
340 *****
341 *****
342 *****
343 *****
344 *****
345 *****
346 *****
347 *****
348 *****
349 *****
350 *****
351 *****
352 *****
353 *****
354 *****
355 *****
356 *****
357 *****
358 *****
359 *****
360 *****
361 *****
362 *****
363 *****
364 *****
365 *****
366 *****
367 *****
368 *****
369 *****
370 *****
371 *****
372 *****
373 *****
374 *****
375 *****
376 *****
377 *****
378 *****
379 *****
380 *****
381 *****
382 *****
383 *****
384 *****
385 *****
386 *****
387 *****
388 *****
389 *****
390 *****
391 *****
392 *****
393 *****
394 *****
395 *****
396 *****
397 *****
398 *****
399 *****
400 *****
401 *****
402 *****
403 *****
404 *****
405 *****
406 *****
407 *****
408 *****
409 *****
410 *****
411 *****
412 *****
413 *****
414 *****
415 *****
416 *****
417 *****
418 *****
419 *****
420 *****
421 *****
422 *****
423 *****
424 *****
425 *****
426 *****
427 *****
428 *****
429 *****
430 *****
431 *****
432 *****
433 *****
434 *****
435 *****
436 *****
437 *****
438 *****
439 *****
440 *****
441 *****
442 *****
443 *****
444 *****
445 *****
446 *****
447 *****
448 *****
449 *****
450 *****
451 *****
452 *****
453 *****
454 *****
455 *****
456 *****
457 *****
458 *****
459 *****
460 *****
461 *****
462 *****
463 *****
464 *****
465 *****
466 *****
467 *****
468 *****
469 *****
470 *****
471 *****
472 *****
473 *****
474 *****
475 *****
476 *****
477 *****
478 *****
479 *****
480 *****
481 *****
482 *****
483 *****
484 *****
485 *****
486 *****
487 *****
488 *****
489 *****
490 *****
491 *****
492 *****
493 *****
494 *****
495 *****
496 *****
497 *****
498 *****
499 *****
500 *****
501 *****
502 *****
503 *****
504 *****
505 *****
506 *****
507 *****
508 *****
509 *****
510 *****
511 *****
512 *****
513 *****
514 *****
515 *****
516 *****
517 *****
518 *****
519 *****
520 *****
521 *****
522 *****
523 *****
524 *****
525 *****
526 *****
527 *****
528 *****
529 *****
530 *****
531 *****
532 *****
533 *****
534 *****
535 *****
536 *****
537 *****
538 *****
539 *****
540 *****
541 *****
542 *****
543 *****
544 *****
545 *****
546 *****
547 *****
548 *****
549 *****
550 *****
551 *****
552 *****
553 *****
554 *****
555 *****
556 *****
557 *****
558 *****
559 *****
560 *****
561 *****
562 *****
563 *****
564 *****
565 *****
566 *****
567 *****
568 *****
569 *****
570 *****
571 *****
572 *****
573 *****
574 *****
575 *****
576 *****
577 *****
578 *****
579 *****
580 *****
581 *****
582 *****
583 *****
584 *****
585 *****
586 *****
587 *****
588 *****
589 *****
590 *****
591 *****
592 *****
593 *****
594 *****
595 *****
596 *****
597 *****
598 *****
599 *****
600 *****
601 *****
602 *****
603 *****
604 *****
605 *****
606 *****
607 *****
608 *****
609 *****
610 *****
611 *****
612 *****
613 *****
614 *****
615 *****
616 *****
617 *****
618 *****
619 *****
620 *****
621 *****
622 *****
623 *****
624 *****
625 *****
626 *****
627 *****
628 *****
629 *****
630 *****
631 *****
632 *****
633 *****
634 *****
635 *****
636 *****
637 *****
638 *****
639 *****
640 *****
641 *****
642 *****
643 *****
644 *****
645 *****
646 *****
647 *****
648 *****
649 *****
650 *****
651 *****
652 *****
653 *****
654 *****
655 *****
656 *****
657 *****
658 *****
659 *****
660 *****
661 *****
662 *****
663 *****
664 *****
665 *****
666 *****
667 *****
668 *****
669 *****
670 *****
671 *****
672 *****
673 *****
674 *****
675 *****
676 *****
677 *****
678 *****
679 *****
680 *****
681 *****
682 *****
683 *****
684 *****
685 *****
686 *****
687 *****
688 *****
689 *****
690 *****
691 *****
692 *****
693 *****
694 *****
695 *****
696 *****
697 *****
698 *****
699 *****
700 *****
701 *****
702 *****
703 *****
704 *****
705 *****
706 *****
707 *****
708 *****
709 *****
710 *****
711 *****
712 *****
713 *****
714 *****
715 *****
716 *****
717 *****
718 *****
719 *****
720 *****
721 *****
722 *****
723 *****
724 *****
725 *****
726 *****
727 *****
728 *****
729 *****
730 *****
731 *****
732 *****
733 *****
734 *****
735 *****
736 *****
737 *****
738 *****
739 *****
740 *****
741 *****
742 *****
743 *****
744 *****
745 *****
746 *****
747 *****
748 *****
749 *****
750 *****
751 *****
752 *****
753 *****
754 *****
755 *****
756 *****
757 *****
758 *****
759 *****
760 *****
761 *****
762 *****
763 *****
764 *****
765 *****
766 *****
767 *****
768 *****
769 *****
770 *****
771 *****
772 *****
773 *****
774 *****
775 *****
776 *****
777 *****
778 *****
779 *****
780 *****
781 *****
782 *****
783 *****
784 *****
785 *****
786 *****
787 *****
788 *****
789 *****
790 *****
791 *****
792 *****
793 *****
794 *****
795 *****
796 *****
797 *****
798 *****
799 *****
800 *****
801 *****
802 *****
803 *****
804 *****
805 *****
806 *****
807 *****
808 *****
809 *****
810 *****
811 *****
812 *****
813 *****
814 *****
815 *****
816 *****
817 *****
818 *****
819 *****
820 *****
821 *****
822 *****
823 *****
824 *****
825 *****
826 *****
827 *****
828 *****
829 *****
830 *****
831 *****
832 *****
833 *****
834 *****
835 *****
836 *****
837 *****
838 *****
839 *****
840 *****
841 *****
842 *****
843 *****
844 *****
845 *****
846 *****
847 *****
848 *****
849 *****
850 *****
851 *****
852 *****
853 *****
854 *****
855 *****
856 *****
857 *****
858 *****
859 *****
860 *****
861 *****
862 *****
863 *****
864 *****
865 *****
866 *****
867 *****
868 *****
869 *****
870 *****
871 *****
872 *****
873 *****
874 *****
875 *****
876 *****
877 *****
878 *****
879 *****
880 *****
881 *****
882 *****
883 *****
884 *****
885 *****
886 *****
887 *****
888 *****
889 *****
890 *****
891 *****
892 *****
893 *****
894 *****
895 *****
896 *****
897 *****
898 *****
899 *****
900 *****
901 *****
902 *****
903 *****
904 *****
905 *****
906 *****
907 *****
908 *****
909 *****
910 *****
911 *****
912 *****
913 *****
914 *****
915 *****
916 *****
917 *****
918 *****
919 *****
920 *****
921 *****
922 *****
923 *****
924 *****
925 *****
926 *****
927 *****
928 *****
929 *****
930 *****
931 *****
932 *****
933 *****
934 *****
935 *****
936 *****
937 *****
938 *****
939 *****
940 *****
941 *****
942 *****
943 *****
944 *****
945 *****
946 *****
947 *****
948 *****
949 *****
950 *****
951 *****
952 *****
953 *****
954 *****
955 *****
956 *****
957 *****
958 *****
959 *****
960 *****
961 *****
962 *****
963 *****
964 *****
965 *****
966 *****
967 *****
968 *****
969 *****
970 *****
971 *****
972 *****
973 *****
974 *****
975 *****
976 *****
977 *****
978 *****
979 *****
980 *****
981 *****
982 *****
983 *****
984 *****
985 *****
986 *****
987 *****
988 *****
989 *****
990 *****
991 *****
992 *****
993 *****
994 *****
995 *****
996 *****
997 *****
998 *****
999 *****
1000 *****

```

```

55 IF INKEY$="B" THEN LET F=1
56 IF F=0 THEN PRINT AT B:1:INK
57 *****: BEEP .0005,50: BEEP .0
58 *****
59 IF ATTR (8,14)=15 OR ATTR
60 (8,15) THEN BEEP .03,20: LET
61 P=0
62 IF ATTR (8,14)=15 THEN BEE
63 P .01,20: LET P=P+20
64 IF P<100 THEN GO TO 400
65 IF P=100 THEN IF INT ((1-33)
66 /200) > 0 THEN PRINT AT 9:1
67 TO FLASH 1:"B O M B U S" FOR I=1
68 TO 10: BEEP .01 THEN BEEP .01
69 NEXT I: LET F=5: LET P=P+(2/RND
70 P)*3
71 *****: PRINT AT 7,11: "
72 *****: PRINT AT 21,3
73 *****: PRINT AT 7,11: INK 6:"S": IN
74 UERSE 1:"TURSE 6:"T": LET T=
75 1:2: LET (1,1)
76 *****: PRINT AT 21,RND*31: INK 7:C
77 H: AND(11+144): RT 21,RND*31:CH
78 130 GO TO 50
79 *****: PRINT AT 3,1: INK 7:C
80 *****: PRINT AT 7,11: INK 2:"M": RT B
81 4:1
82 *****: PRINT AT 5,11: INK 6:"ODU":
83 *****: PRINT AT 5,11: INK 6:"ODU":
84 *****: BEEP .0005,60: NEXT J
85 *****: NEXT N
86 *****: PRINT AT 8,11
87 *****: IF B=5 THEN LET B=13
88 *****: INK 7: IF e(1)=B THEN GO T
89 O 500
90 *****: LET F=1: LET e(1)=B:
91 *****: PRINT AT 8,11: SUPERIO "T
92 *****: FOR PUNTEGGIO RECORD 1:
93 *****: FOR I=1 TO 10: I=1 TO 6
94 *****: BEEP .005,100: NEXT I: NEXT J
95 *****: PRINT LE INIZIALI DEL
96 *****: LET N=8
97 *****: NEXT I: PRINT N: PR
98 *****: CODE INKEY$: LET ANS=INKE
99 *****: NEXT I: PRINT N: PR
100 *****: PRINT AT 4,0: PUNTEGGIO "
101 *****: BEEP .01: PUNTEGGIO "B
102 *****: RECORD ? FLASH 1:eti1: " di
103 *****:
104 *****: PRINT AT 14,0:"premi
105 *****: il tasto per registrare il tuo record"
106 *****: PRINT "ENTER per un'altra
107 *****: partita"

```

```

550 PAUSE 0
551 IF CODE INKEY$=13 THEN GO T
552 O 500
553 IF CODE INKEY$=11 THEN GO
554 TO 500
555 GO TO 550
556 SAVE NS DATA e(1)
557 LET (1,1)
558 FOR #1 TO 21
559 RESTORE
560 FOR #2 TO 7
561 READ P#
562 READ a: PDKE USE P#*#*#
563 NEXT #
564 DATA "a",52,33,255,129,129,
565 37,25
566 DATA "b",112,140,130,65,129
567 129,24,32
568 DATA "c",0,0,24,36,36,50,0,0
569 0,0
570 DATA "d",0,0,48,40,48,32,0,0
571 0,0
572 DATA "e",0,24,40,16,0,0,0,0
573 0,0
574 DATA "f",112,136,136,66,40,
575 48,0,0
576 DATA "g",0,0,0,56,44,16,0,0,0
577 0,0
578 DATA "h",0,0,30,16,34,64,23
579 0,0
580 DATA "i",94,177,130,228,34,
581 55,69,50
582 DATA "j",0,62,65,66,34,17,1
583 4,0
584 DATA "k",95,144,144,116,10,
585 49,65,126
586 DATA "l",62,65,242,9,247,13
587 1,0
588 DATA "m",145,62,16,7,244,8,0
589 4,197
590 DATA "n",100,24,195,0,222,0,
591 37,137
592 DATA "o",0,4,34,18,1,68,50,0
593 0,0
594 DATA "p",95,60,1,18,34,4,4,0
595 0,0
596 DATA "q",0,8,170,42,73,65,1,0
597 0,0
598 DATA "r",144,76,34,128,72,6
599 0,0
600 DATA "s",0,3,15,15,120,200,0
601 0,0
602 DATA "t",0,192,240,240,30,1
603 2,25,28
604 DATA "u",0,32,68,72,128,36,
605 76,144

```

# VIC 20

comodore



## a casa vostra subito!

Se volete riceverlo velocemente compilate e spedite in busta il "Coupon VIC 20"

**EXELCO**  
Via G. Verdi, 23/25  
20095 - CUSANO MILANINO (MILANO)

Descrizione	Qt.	Prezzo unitario	Totale L.
Personal Computer VIC20		L. 199.000	
Registatore a cassetta C2N-VIC1530		L. 110.000	
Cartridge di espansione 8K RAM-VC1110		L. 95.000	
Cartridge di espansione 16K RAM-VC1111		L. 125.000	
Espansione per alta risoluzione 3 KB - VC1211N		L. 75.000	
Floppy Disk VC1541		L. 585.000	
Stampante SEIKOSHA-GP100VC		L. 550.000	
Joystick - VC1311 - singolo		L. 10.000	
Paddle - VC 1312 - la coppia		L. 20.000	
Impariamo a programmare in Basic con il VIC20		L. 9.000	
Guida al Personal Computer VIC20		L. 20.000	

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

Nome

Cognome

Via

Città

Data  C.A.P.

Desidero ricevere la fattura  SI  NO

Partita I.V.A. o, per i privati Codice Fiscale

Acconto L.

Sarà data precedenza alle spedizioni, se assieme all'ordine verrà incluso un anticipo di almeno L. 10.000. Aggiungere L. 5.000 per il recapito a domicilio. prezzi vanno maggiorati dell'I.V.A. 18%

# “Wei-ch'i”, un gioco cinese per PET

Un antico gioco di strategia, che sicuramente vi appassionerà

di Umberto Barzagli

**I**l Wei-ch'i (che i giapponesi chiamano GO, nome con cui è più noto in occidente), è un gioco da tavolino di strategia per due persone e, come per gli scacchi, fortuna e destrezza fisica non hanno alcuna influenza sul gioco.

La scacchiera è rappresentata da una semplice grata quadrata, di solito costituita da 19 linee orizzontali e 19 verticali, equidistanti tra loro; la scacchiera del tipo più diffuso (è riprodotta su piatti, vassoi e tavolini e, anche nel recente “Marco Polo” televisivo c'era una scena in cui il Gran Kan dei tartari gioca una partita con il figlio), è quindi composta da 361 intersezioni. Nove di queste vengono indicate convenzionalmente, per servire come punti di riferimento. Si adoperano anche altre scacchiere: le più comuni sono quelle 8 x 8 (con quattro caselle di riferimento, invece di nove), 13 x 13 e 17 x 17.

I teorici del Wei-ch'i affermano che le combinazioni strategiche per una scacchiera 23 x 23 o ancora maggiore, andrebbero al di là delle possibilità della mente umana. Una scacchiera più grande produce un aumento delle complicazioni strategiche in progressione geometrica. Se, quindi, il gioco su una scacchiera

8 x 8 è semplice e quasi soltanto tattico, il gioco su scacchiere di 17 e più intersezioni per lato assume aspetti prettamente strategici e di più ampio respiro rispetto al confronto spicciolo tra poche pedine.

Il Wei-ch'i si gioca con pezzi bianchi e neri che vengono chiamati “pietre” o “uomini”. All'inizio del gioco la scacchiera è vuota, a meno che non sia stato concesso un handicap. I bianchi ed i neri giocano con mosse alternate; il nero ha la prima mossa. Ad ogni mossa si colloca un pezzo su una intersezione vacante. Il numero dei pezzi assegnati ad ogni giocatore è pari al numero totale delle intersezioni diviso due, più una eventuale pedina aggiuntiva per il nero se si gioca con scacchiere con un numero dispari di intersezioni per lato - nel caso più diffuso di una scacchiera 19 x 19, abbiamo 181 pedine nere e 180 bianche. Una volta giocato, il pezzo non può più essere mosso (se non rimosso in caso di cattura).

Quando vi sia una marcata differenza di abilità tra i due giocatori, il giocatore più debole può ottenere un handicap. Contrariamente al significato “occidentale” del termine “handicap”, nel Wei-ch'i esso indica un vantaggio artificiale concesso al giocatore di livello inferiore, che ottiene i pezzi neri, ne colloca un certo numero sulla scacchiera, secondo un disegno determinato prima dell'inizio del gioco, e lascia quindi al giocatore più forte le pedine bianche ed il diritto di fare la prima mossa. Il numero dei pezzi “handicap”, varia a seconda della differenza di abilità tra i due giocatori.

In pratica il giocatore con handicap dispone di solito di un numero di pedine “handicap” che va da un minimo di due ad un massimo di

nove. Le pedine possono essere disposte comunque, ma vengono di solito piazzate su parte o anche su tutte le intersezioni di riferimento.

Quando due o più pezzi di un colore sono collocati in posizione da formare una connessione a catena, vengono considerati una singola unità e chiamati “gruppo”. I pezzi si considerano collegati fra loro quando si uniscono su una linea, ma non se sono uniti l'un l'altro diagonalmente.

Due sono gli obiettivi del Wei-ch'i: il controllo del territorio e la cattura dei pezzi nemici. Questi due obiettivi sono strettamente connessi nella strategia del gioco, dato che la cattura di pezzi nemici porta, di solito, alla conquista di nuovo territorio, e la costruzione di basi territoriali provoca, spesso, ma non necessariamente, ulteriori catture.

In generale una parte può catturare un pezzo nemico o un gruppo giocando i suoi pezzi su tutte le intersezioni subito adiacenti. I singoli pezzi di un gruppo possono essere presi solo per cattura, e cioè con l'accerchiamento di tutto il gruppo.

Se un pezzo o un gruppo è situato in maniera da toccare uno o più punti lungo i lati della scacchiera, per catturarlo basta accerchiarlo dalla parte interna. Se viene attaccato un gruppo che circonda una o più intersezioni la sua cattura da parte dell'avversario avviene solo con l'accerchiamento sia all'interno che all'esterno.

Alla fine di una partita di Wei-ch'i, tutto il territorio possibile è circondato dalle pedine dell'uno o dell'altro giocatore, e ambedue i contendenti hanno catturato e “ucciso” tutti i pezzi possibili. Il gioco termina per accordo di entrambe le parti, e quando le “linee” sono ben determinate.

## “Wei-ch'i”, un gioco cinese per PET

### Wei-ch'i, il programma

Il programma sviluppato per PET serie 32, (3032 o 4032), deve, purtroppo, sottostare ad alcune gravose limitazioni dovute alla complessità del gioco. La limitazione più rilevante è data dalla impossibilità di realizzare una subroutine che consenta una analisi esaustiva e, quel che più conta, rapida dei casi di cattura.

Dato infatti che i possibili casi di accerchiamento sono molto numerosi, una subroutine che riuscisse ad essere esaustiva sarebbe comunque troppo lenta per poter essere inserita nell'economia del programma.

Per questa ragione sono state realizzate due “release” del programma; quello che appare sulla rivista è la seconda delle due. La differenza rispetto all'originale è data dalla presenza nella seconda di una subroutine che esaurisce i casi di cattura, quando sono limitati ad una sola pedina. La pesantezza di questa analisi può essere facilmente rilevata osservando come aumenta la velocità di gioco sopprimendo la riga 199 che accede alla coppia di routine che realizza l'analisi suddetta.

Nella prima “release”, che non comprendeva le subroutine da 17000 a 20000, la determinazione dello stato di cattura veniva lasciata interamente al giocatore. Nella seconda “release” il giocatore, dopo ogni pedina giocata, esaurisce l'analisi delle catture singole (determinata ovviamente dall'ultima pedina giocata), lasciando ai giocatori la determinazione della cattura di “gruppi”.

I due giocatori possono rimuovere pedine dalla scacchiera, premendo, in risposta alla domanda del calcolatore circa in quale colonna si vuole fare la prossima mossa, il tasto corrispondente alla frecciolina

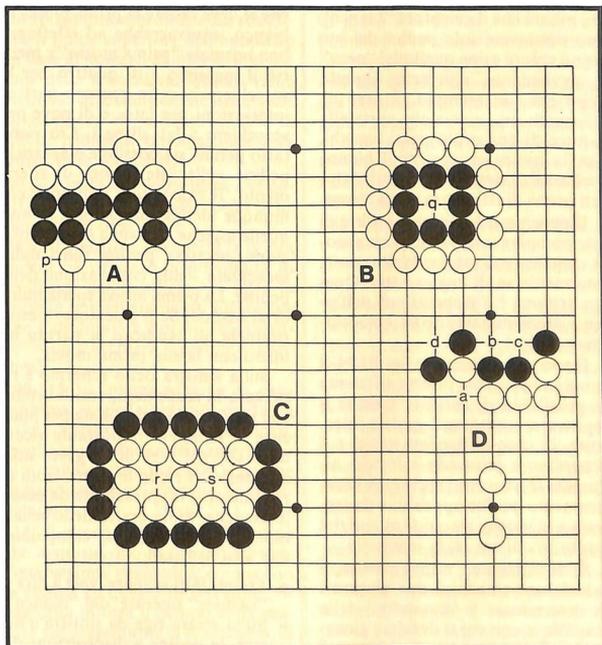


Figura 1. A) Il gruppo nero (nove pezzi) a contatto con il lato sinistro della scacchiera, può essere catturato dal bianco se questi gioca in “p”.

B) Il gruppo nero (otto pezzi) che circonda l'intersezione “q”, per poter essere catturato deve essere circondato sia all'interno che all'esterno. Il bianco deve dunque giocare in “q” per poter catturare il gruppo nero.

C) Il gruppo bianco (tredici pezzi), pur circondato esternamente dal nero, non può più essere catturato da questi. Se il nero gioca in “r”, non si verificherà l'accerchiamento del gruppo dei bianchi (liberi in “s”), ma il pezzo nero giocato in “r” sarà circondato dai pezzi bianchi adiacenti e catturato. Analoga sarebbe la conseguenza se il nero giocasse in “s”.

D) Un esempio classico a livello tattico. Se la mossa spetta al nero, potrebbe giocare in “a” catturando la pedina bianca, aggirando la posizione dell'avversario e riducendone l'influenza.

Supponiamo che tocchi al bianco muovere per primo. Se gioca in “a”, può costringere il nero a legare in “b” o in “c” e tagliare in due le forze nere. E dopo che il nero ha mosso in “b”, il bianco può giocare in “d”, colpendo così un altro punto debole della posizione del nero.

**“Wei-ch’i”,**

**— un gioco cinese per PET**

verso l'alto, ed indicando, quindi, quante pedine si vogliono rimuovere. Il calcolatore non può, ovviamente, verificare che queste “rimozioni” siano legali ai fini del gioco. Da notare che il giocatore “bianco” può rimuovere solo pedine del suo stesso colore e non quelle del “nero” e, ovviamente, viceversa. Quindi, dopo che, ad esempio, il nero ha effettuato una mossa che porta alla cattura di due o più pezzi bianchi, con la mossa successiva il bianco avrà cura di rimuovere i pezzi catturati prima di effettuare la sua mossa.

Il programma consente inoltre di interrompere una partita non ancora conclusa e di salvare la situazione intermedia su di una cassetta. Questa opzione è a disposizione dell'utente premendo il tasto @ (chiocciolina).

Dopo aver mandato in RUN il programma, viene offerta all'utente la possibilità di caricare tramite il registratore eventuali partite interrotte. In caso di risposta positiva il programma provvede a fornire all'utente le istruzioni necessarie al caricamento del file opportuno da cassetta e la partita riprende quindi dal punto in cui era stata interrotta.

Se si inizia una nuova partita, il calcolatore chiede ai due giocatori di determinare le dimensioni della scacchiera con cui si desidera giocare. Le dimensioni (viene richiesto il numero di intersezioni per lato) sono limitate per ragioni logiche ad un minimo di 8, con scacchiere con un numero di intersezioni per lato inferiore, il gioco perde praticamente di significato; e per ragioni grafiche ad un massimo di 23.

Dopo aver inizializzato le matrici necessarie allo svolgimento del gioco (il tempo di attesa è quindi proporzionale al numero di intersezioni della scacchiera prescelta); il calcolatore richiede al giocatore “nero” di indicare premendo i tasti S o N, se desidera disporre di un handicap.

In caso di risposta affermativa, il giocatore chiede al “nero” di indica-

re quanti pezzi “handicap” desidera piazzare sulla scacchiera.

Il minimo è fissato in due pedine per ogni tipo di scacchiera (piazzarne una sola non avrebbe senso, dato che si deve cedere la prima mossa al bianco, equivarrebbe ad effettuare una normale “prima mossa”); mentre il massimo è di quattro per le scacchiere con un numero pari di intersezioni per lato, e di nove per scacchiere a lati dispari. Ciò viene fatto perché sia possibile piazzare le pedine sulle intersezioni di riferimento. Il calcolatore non pone comunque alcuna limitazione, se non quelle logiche - nessuna sovrapposizione, nessuna pedina fuori dalla scacchiera - alla collocazione delle pedine. La prima mossa spetta quindi al bianco. Se il giocatore “nero” rinuncia all'handicap la partita ha inizio con la sua prima mossa.

Sulla sinistra dello schermo è riportata la scacchiera, con le lettere che servono da coordinate per piazzare le mosse (è importante ricordarsi che nel Wei-ch’i si gioca sulle intersezioni) e con le intersezioni di riferimento, rappresentate da caselle bianche. Sulla destra dello schermo, su fondo bianco, compaiono dall'alto in basso:

- I punti fatti segnare, vale a dire le “catture” operate, dal “bianco”.
- Sulla stessa riga da sinistra a destra, le pedine a disposizione del bianco e le pedine giocate.
- Il tempo di gioco impiegato dal “bianco” per pensare le proprie mosse in ore, minuti primi e minuti secondi.
- Il numero di mosse effettuate da entrambi i giocatori.
- Il tempo totale di gioco.
- Le prime tre voci dell'elenco per il giocatore “nero”.

Ai piedi di questo tabellone comparirà, di volta in volta la scritta che invita ad introdurre le coordinate di colonna e di riga per effettuare ogni nuova mossa. Il giocatore cui tocca muovere è identificato dal colore di fondo della scritta: se i termini CO-

LONNA e RIGA, compaiono in forma normale (bianco su nero), la mossa spetta al giocatore “nero”; se compaiono in forma inversa (“reverse” o nero su bianco), la mossa tocca al bianco.

Oltre alla possibilità di eliminare le pedine dalla scacchiera e di salvare delle partite in corso di svolgimento si è ritenuto opportuno offrire una ulteriore opzione ai giocatori. Grazie ad un buffer, limitato, per ragioni di occupazione di memoria a sole dieci mosse, è possibile risalire a punti precedenti della partita per riprenderla in momenti cruciali della partita stessa. Per utilizzare questa opzione è sufficiente premere il tasto “frecciolina verso sinistra” e, alla domanda del calcolatore, indicare quante delle ultime mosse si desidera annullare, con la logica limitazione che il numero non sia superiore all'ampiezza del buffer. Il calcolatore registra le mosse effettuate e, quindi, considera nel conto delle ultime n mosse, anche quelle che sono state successivamente rimosse perché catturate. Non è ovviamente possibile usufruire più volte di seguito della opzione suddetta se si è arrivati ad esaurire il buffer delle mosse precedenti.

Il calcolatore provvede ovviamente a segnalare tutti i casi di errore, come sovrapposizioni o pedine mal collocate.

**0-47** Acquisizione dei dati necessari per l'inizializzazione del programma. In particolare:

**0** Dimensionamento del vettore gestione-video e inizializzazione delle variabili:

H - numero di pedine di handicap.  
M - giocatore a cui spetta la mossa (la variabile M è inizializzata a -1, poichè spetta al giocatore “nero” iniziare la partita). PB\$, punti spettanti al giocatore “bianco”.

**3** Inizializzazione di altre variabili utilizzate all'interno del programma: PN\$, analogo a PB\$, ma per il giocatore “nero”; BH\$, tempo impiegato per pensare le proprie mosse

**“Wei-ch’i”,  
un gioco cinese per PET**

da parte del giocatore “bianco”; NH\$, analogamente per il “nero”; MO\$, numero di mosse compiute in totale dai due giocatori nel corso della partita; BU, ampiezza del “buffer” che contiene le ultime (BU) mosse in modo che la partita possa essere sospesa e ripresa da un numero di mosse precedenti quella in corso minore o uguale a BU.

**4** Ulteriore inizializzazione di variabili.

**7-8** Viene data all’utente l’opportunità di ri-iniziare una partita, precedentemente interrotta.

**10-20** Nel caso in cui l’utente abbia desiderato iniziare una nuova partita, viene richiesta l’introduzione del numero di intersezioni per lato della scacchiera con cui desidera giocare. Le intersezioni sono state inferiormente limitate al numero di 8 e superiormente a 23 poiché queste sono le dimensioni “limite” tradizionalmente accettate dai Wei-ch’i.

**25** Dimensionamento di alcune matrici necessarie all’interno del programma. In particolare: S(L, L), matrice rappresentante il contenuto della scacchiera; X(BU) e Y(BU), vettori contenenti le coordinate delle ultime BU pedine giocate; M(BU), vettore contenente le mosse delle ultime BU pedine.

**30** Vengono inizializzate le variabili contenenti le pedine a disposizione del “bianco” e del “nero” all’inizio della partita.

L’attribuzione delle variabili è fatta in modo che la somma delle pedine “nere” e di quelle “bianche”, all’inizio della partita, sia pari al numero di intersezioni della scacchiera. Nel caso in cui il numero di queste intersezioni sia dispari, al giocatore “nero”, che è il giocatore destinato ad iniziare la partita, viene attribuita una pedina in più.

Inoltre viene chiesto all’utente se desidera usufruire o no di un handicap.

N.B. L’handicap è a disposizione del giocatore “nero”, che inizia la partita, il quale, dopo aver piazzato sulla

scacchiera un numero di pedine pari all’handicap pattuito, lascia la prima mossa al “bianco”.

**45** Si accede, nel caso in cui il “nero” abbia scelto di usufruire di un handicap, ad una subroutine che stabilisce il limite superiore dell’handicap in base al numero di intersezioni per lato della scacchiera. Quindi viene richiesto al giocatore che usufruisce di handicap di indicare quante di queste pedine di vantaggio desidera porre sulla scacchiera.

**50** Si accede alla subroutine che costruisce sul video la scacchiera delle dimensioni desiderate.

**94** Se il giocatore cui spetta la prima mossa ha rifiutato la possibilità di usufruire dell’handicap, si salta la sezione immediatamente successiva che è destinata a porre “fisicamente” sulla scacchiera le pedine di handicap.

**100-115** In questa sezione di programma il giocatore cui spetta la prima mossa e utilizza l’handicap, sceglie le coordinate delle intersezioni su cui desidera piazzare le pedine di handicap di cui dispone. In particolare:

**110** Si accede alle due subroutine che acquisiscono le coordinate dell’intersezione desiderata e pongono la pedina su questa intersezione. Si aggiorna inoltre il campo della matrice rappresentante la scacchiera, di indici pari alle coordinate della mossa effettuata, con un valore numerico pari alla variabile M, che indica a quale dei due giocatori spetta la mossa. Se M è uguale a -1, la mossa spetta (come in questo caso) al nero, se M è uguale a +1, la mossa tocca al bianco.

Pertanto, se sulla scacchiera la posizione di indici X, Y contiene una pedina nera, nella matrice S (X, Y), la posizione di indici corrispondenti, sarà pari a -1, se la pedina è bianca la casella di indici X, Y sarà uguale a +1; se nella posizione suddetta non ci fosse alcuna pedina, il valore della matrice S corrispondente il valore numerico sarebbe uguale

a zero.

**115** Terminato il ciclo di introduzione delle pedine di handicap la mossa viene “passata” (M = 1) al bianco.

**130-145** Viene riempito il tabellone di gioco alla destra dello schermo contenente la situazione evolutasi nel corso della partita.

**150-200** Questa sezione di programma costituisce il nocciolo del programma stesso e funge da “sezione di collegamento e smistamento” tra le varie subroutine che vengono così richiamate nella sequenza opportuna. In particolare:

**150** Viene azzerato il tempo di gioco e si accede alla subroutine che aggiorna le variabili contenute nel tabellone di gioco.

**160** Nel caso in cui i due giocatori siano rimasti senza pedine a disposizione si accede alla subroutine che segnala la conclusione della partita.

**170** Si accede alla subroutine che acquisisce le coordinate della intersezione in cui si desidera porre la propria pedina e, in seguito alla subroutine che provvede ad aggiornare i tre orologi presenti sul tabellone di gioco.

**175** Si accede alla subroutine che piazza sullo schermo la pedina giocata e si aggiorna il valore delle variabili MO (contenente il numero di mosse effettuate) e la posizione di indici pari alle coordinate dell’ultima mossa effettuata all’interno della matrice rappresentante la scacchiera.

**180-190** Queste due righe sono analoghe come funzione e contenuto. Entrambe, dopo aver richiamato la subroutine che aggiorna gli orologi, aggiornano le variabili contenenti il numero di pedine a disposizione del “nero” o del “bianco”, a seconda di quale dei due ha effettuato l’ultima mossa.

**197-198** Si accede alla subroutine che provvede ad elencare nell’apposito buffer le coordinate dell’ultima mossa effettuata e quale dei due giocatori ha operato la stessa.

**199** Si accede a due subroutine che,

**"Wei-ch'i",  
un gioco cinese per PET**

in successione, compiono un esame della situazione di gioco per verificare se l'ultima pedina giocata ha contribuito a circondare una (ed una sola!) pedina avversaria, o, se la pedina giocata è stata "sacrificata" dall'autore dell'ultima mossa. Poiché queste due subroutine vengono richiamate dopo ogni mossa, la loro influenza sul tempo di gioco è rilevante. Se lo si desidera, si può sopprimere l'intera riga; in questo modo, il compito di verificare gli eventuali accerchiamenti, ricade interamente sul giocatore.

**200** Si accede alla subroutine che aggiorna la situazione generale del tabellone, per quel che riguarda mosse, punti acquisiti, pedine giocate e pedine disponibili. Quindi si accede alla subroutine che aggiorna gli orologi del tabellone stesso. Infine si cambia il valore della variabile M nel suo opposto in modo da passare la mossa al giocatore avversario e quindi si ricicla alla linea 160 in cui viene ripresa l'esecuzione del nocciolo del programma.

**550-592** Questa subroutine provvede a mostrare sul video la scacchiera, costruendola in base alle specifiche determinate dai dati introdotti di volta in volta dall'utente. In particolare:

**550-580** Vengono creati i quattro bordi della scacchiera correndoli con le lettere che i giocatori devono utilizzare per indicare le coordinate del punto in cui si vuole effettuare una mossa.

**590** Viene riempita la scacchiera con le intersezioni.

**591** Dopo aver provveduto ad indicare le lettere sul bordo inferiore della scacchiera, si accede alla subroutine che provvede a piazzare delle caselle di riferimento per identificare le posizioni chiave della scacchiera in cui, generalmente (ma non vi è alcuna regola tassativa), vengono piazzate le pedine di handicap. Viene inoltre determinato il valore delle variabili L0 e L6, utilizzate per posizionare opportunamente il ta-

bellone di gioco sul video, rispetto alla scacchiera.

**592** Viene realizzata la base del tabellone di gioco.

**1000-1520** Questa subroutine provvede a piazzare le caselle handicap sulla scacchiera. Essa è divisa in due sezioni cui il calcolatore accede a seconda che il numero di intersezioni per lato sia pari o dispari. In particolare:

**1000-1030** In riga 1000 viene effettuato il test sulla parità o disparità della variabile L, rappresentante il numero di intersezioni per lato. Se la risposta è positiva si salta alla sezione da 1500 a 1520. Da 1010 a 1030 vengono posizionate le caselle di riferimento per gli handicap nel caso in cui la scacchiera abbia un numero dispari di intersezioni per lato.

**1010** Il numero di caselle handicap è in questo caso uguale a nove. K è l'indice che seleziona in base alle dimensioni della scacchiera prescelta, una delle otto righe seguenti, contenenti le variabili che identificano i limiti ed il passo della configurazione particolare per la scacchiera delle dimensioni volute delle caselle handicap.

**1020-1030** Queste righe, con un ciclo doppio, provvedono a piazzare fisicamente le caselle handicap sulla scacchiera.

**2000-2130** Subroutine responsabile della acquisizione da tastiera dei caratteri introdotti dai due giocatori, sia per indicare le coordinate del punto della scacchiera in cui si vuole piazzare una pedina, sia per sfruttare una delle opzioni offerte dal programma nel corso del gioco.

In particolare:

**2000-2010** Display della richiesta di introduzione di un carattere indicante l'indice di colonna delle coordinate richieste. Si accede inoltre alla subroutine 2100, responsabile della acquisizione dei caratteri da tastiera.

**2015-2020** Test di congruenza nel caso in cui si sia utilizzata una delle opzioni di cui sopra.

**2025-2027** Viene ricavato dal carattere introdotto un valore numerico identificante gli indici corrispondenti al punto voluto della scacchiera nella matrice che la rappresenta all'interno del programma.

**2030-2040** Analogamente alle righe precedenti per quanto riguarda l'indice di riga.

**2100-2130** Viene "fisicamente" acquisito da tastiera il carattere voluto. In particolare:

**2110** Se il carattere introdotto è uguale a @ (chiocciolina), si accede alla subroutine che permette di salvare su nastro una partita in corso di svolgimento, in modo che possa essere ripresa dal punto in cui è stata interrotta.

**2120** Si accede, in base al carattere introdotto (1 frecciolina verso l'alto), alla subroutine che permette di rimuovere dalla scacchiera le pedine "circondate".

**2130** Se il carattere introdotto corrisponde alla frecciolina verso sinistra (←), si accede alla subroutine che permette di riportare la situazione ad un numero di mosse prima dell'attuale, minore o uguale all'ampiezza del buffer che il computer dedica a questo scopo (nel nostro caso BU è stata inizializzata a 10 in riga 3; se si desidera l'ampiezza del buffer può essere aumentata, ma ciò porta a due inconvenienti: innanzi tutto ogni mossa sarà rallentata dalla necessità di shiftare tutte le mosse contenute nel buffer per fare posto all'ultima, una volta che il buffer sia pieno; in secondo luogo è difficile far riferimento ad un punto preciso della partita, quando si deve effettuare un salto all'indietro, per raggiungerlo, che sia superiore ad una mezza dozzina di mosse).

**3000-3050** Questa subroutine piazza sulla scacchiera la pedina nella posizione voluta, operando contemporaneamente un controllo sulla correttezza della giocata. In particolare:

**3000** Se la intersezione su cui si desidera porre la propria pedina risulta

## “Wei-ch'i”, un gioco cinese per PET

già occupata, viene segnalato l'errore.

**3020-3030** Se la intersezione è uno dei nodi di riferimento per le pedine di handicap, il numero di POKE corrispondente al carattere da mostrare sul video, viene modificato in modo da ottenere una uscita congruente (una semplice uscita in negativo del carattere normale non sarebbe infatti esatta, poiché una pedina bianca (CR = 81), in negativo (CR = 209), diviene, ovviamente, una pedina nera su fondo bianco).

**3040-3050** Mediante un ciclo viene realizzato il lampeggio.

**4000-4040** Subroutine di salvataggio su nastro di una partita in corso.

**5000-5040** Questa subroutine permette di eliminare dalla scacchiera un numero qualsiasi di pedine dello stesso colore. Il suo uso più logico consiste, ovviamente, nell'utilizzarla per rimuovere dalla scacchiera le pedine “circondate”, ma può anche servire per creare delle situazioni particolari su cui compiere esercizi di tattica o per rimediare ad errori di qualsiasi genere. Dato che l'introduzione di questa subroutine si è resa necessaria a causa della impossibilità di realizzare una funzione esauritiva e rapida che verificasse l'eventuale cattura di due o più pedine, non vi è alcun controllo sulla legittimità delle operazioni realizzate tramite questo sottoprogramma, ed esso va quindi usato con attenzione per non provocare un irregolare evolversi della partita.

N.B. Come detto possono essere rimosse solo le pedine del giocatore cui spetta la mossa (quindi anche tutte quelle del suo colore, che si venissero eventualmente a trovare sulla scacchiera), il computer provvede a segnalare ogni sorta di errore al riguardo.

L'identificazione della pedina che si desidera rimuovere è in tutto e per tutto simile ad una normale mossa all'interno del gioco: il computer richiede la lettera corrispondente alla colonna su cui la pedina si trova,

seguita dall'indice di riga della stessa. Va fatta ovviamente molta attenzione per non compiere operazioni illecite.

**6000** Subroutine che provvede a cancellare determinate aree del video in cui compaiono le scritte.

**7000-7130** Questo sottoprogramma provvede a caricare da nastro ed a ripristinare la situazione antecedente ad una interruzione nel corso di una partita. In particolare:

**7000-7040** I dati vengono riacquisiti da nastro nello stesso ordine in cui sono stati salvati.

**7033** Vengono dimensionate le matrici utilizzate all'interno del programma, in base ai limiti caricati da nastro.

**7050-7080** Viene ripristinato il tabellone di gioco in base ai dati letti.

**7085** Si accede alla subroutine che “disegna” la scacchiera ed il tabellone di gioco.

**7090-7130** Viene ricostruita la situazione antecedente all'interruzione, quindi viene ceduta la mossa al giocatore a cui spetta.

**9000-9920** Questa lunga e, apparentemente, complicata subroutine provvede a tenere aggiornati i tre orologi che si trovano sul tabellone di gioco. Nel corso del gioco non vi sono mai più di due orologi in funzione: uno è quello centrale, che misura la durata totale della partita, l'altro è quello dei due orologi restanti che misura il tempo per cui il giocatore cui tocca muovere, pensa le proprie giocate.

**11000-11080** Questo sottoprogramma viene richiamato dalla subroutine che provvede a rimuovere le pedine dalla scacchiera una volta che siano state catturate. Essa provvede a determinare il carattere che va sostituito alla pedina rimossa in base alla posizione della stessa sulla scacchiera. In particolare:

**11000** Alla variabile CR viene assegnato il valore numerico 91, corrispondente al numero di POKE di una normale intersezione.

**11000-11030** Viene eseguito un test

tendente a determinare se la posizione della pedina da eliminare è su uno dei quattro lati della scacchiera. Se il test è positivo, si provvede a sostituire il carattere intersezione normale (CR = 91), con un valore opportuno.

**11040-11070** Viene operato un test analogo a quello delle righe precedenti per i quattro angoli della scacchiera.

**11073** Si determina con un ulteriore test se la pedina occupava una delle caselle di riferimento per gli handicap.

**11075-11080** Viene piazzato sulla scacchiera il carattere opportuno, cancellando, così, la pedina da rimuovere.

**12000-12510** Questa subroutine provvede ad aggiornare, sul tabellone di gioco, la situazione di punteggio dei due giocatori dopo ogni cattura.

**13000-13010** Sottoprogramma di aggiornamento del buffer contenente le ultime BU mosse. In particolare:

**13000** Se il puntatore MO alla cima del buffer supera il limite superiore, si accede ad un ulteriore sottoprogramma che provvede a shiftare l'intero buffer per creare lo spazio ad un ulteriore inserimento.

**14000** Sottoprogramma che provvede a shiftare il buffer di una posizione; ovviamente il contenuto del buffer corrispondente alla mossa attuale - 10, viene perduto.

**15000-15020** Questa subroutine consente di effettuare un “salto all'indietro” nell'andamento della partita, la cui ampiezza dipende dall'ampiezza del buffer (in riga 3, la variabile BU, che rappresenta l'ampiezza del buffer è inizializzata a 10). Ovviamente si può ricorrere a questo sottoprogramma un numero consecutivo di volte limitato, onde non vuotare completamente il buffer del suo contenuto. Si consiglia quindi di utilizzare moderatamente la opzione suddetta e di porre molta

attenzione nell'indicare di quante

**"Wei-ch'i",  
un gioco cinese per PET**

mosse si vuole tornare indietro. In particolare:

**15010** Una volta acquisita da tastiera l'ampiezza del "salto", il computer ricicla nel caso in cui il salto richiesto superi l'attuale livello di riempimento del buffer.

**15015-15020** Mediante un ciclo si provvede a richiamare ordinatamente, per ognuna delle ultime MB mosse, le subroutine responsabili della rimozione delle pedine giocate e dell'aggiornamento della situazione.

**16000-16500** Questo sottoprogramma, ausiliario del precedente, aggiorna il numero di pedine giocate e da giocare per entrambi i giocatori, restituendo loro quelle impegnate nelle mosse che vengono soppresse dalla subroutine 15000.

**17000-17900** Questa subroutine esamina il caso in cui l'ultima pedina giocata abbia contribuito alla cattura di una singola pedina avversaria. In particolare:

**17000-17003** I due cicli annidati permettono di esaminare le pedine circostanti l'ultima pedina giocata.

**17005** Il test consente di escludere dall'esame precedente sia la pedina in questione, che le pedine adiacenti a quella giocata per ultima ma in posizione angolata rispetto a questa, che nel Wei-ch'i non influenzano la cattura della pedina avversaria, né ne vengono influenzate.

**17010** Si accede al sottoprogramma che provvede a compiere l'esame su ogni singola pedina.

**17800** Questa subroutine provvede a richiamare i sottoprogrammi che rimuovono la pedina catturata dallo schermo ed aggiornano la situazione di punteggio e la scacchiera "logica" S(X,Y).

**18000-18010** Questo sottoprogramma diminuisce l'indice di accerchiamento TC, nel caso in cui la pedina che si sta esaminando appartenga ad una intersezione di lato o di angolo, in questo caso, infatti, l'aggressore può avvalersi di un effetto di "sponda", sfruttando il bordo - o i bordi -

della scacchiera come se fossero parte integrante del proprio schieramento.

**19000-19120** Questa subroutine è responsabile della determinazione della situazione di accerchiamento per le singole pedine. In particolare: **19000-19010** I due cicli annidati permettono di esaminare le pedine circostanti (quelle responsabili della cattura per la pedina in esame). Il test in 19010 consente di escludere dal test la pedina in esame.

**19020** Il test esclude le pedine "angolate" rispetto alla pedina in esame.

**19030** Il test evita dei richiami "illelegali" a caselle al di fuori dei limiti della scacchiera.

**19035** Se la casella adiacente a quella in esame è vuota, la variabile CT, che rappresenta il livello "reale" di accerchiamento, non viene aumentata, segnalando così che la pedina in esame non è attualmente circondata.

**19040** Se la casella adiacente è occupata da una pedina avversaria, la variabile CT viene aumentata di una unità.

**19110** Se il livello attuale di accerchiamento è pari a quello necessario

per far ritenere la pedina accerchiata si accede alla subroutine che provvede a rimuovere la pedina catturata e ad aggiornare il tabellone di gioco. **20000** Questo sottoprogramma consente di inizializzare opportunamente le variabili X0 e Y0, prima di accedere alla subroutine che esamina lo stato di cattura delle pedine, in modo da esaminare la possibilità che l'ultima pedina giocata sia stata posta, per errore, in una casella già circondata dalle pedine avversarie. Come si può notare da riga 199, si accede a questa subroutine, solo dopo aver esaminato le pedine circostanti all'ultima pedina giocata, in modo da consentire la cattura di pedine per cui si renda necessario un accerchiamento sia interno che esterno. Se, infatti, l'esame per l'ultima pedina giocata venisse eseguito prima - o nell'ambito dello stesso ciclo - dell'esame delle pedine circostanti, verrebbero scorrettamente rimosse anche quelle pedine che erano invece state giocate con l'intento di completare l'accerchiamento dall'interno dello schieramento avversario. Per maggior chiarezza, vedi il diagramma B di figura 1.

Figura 2. Il listato BASIC.

```

0 CLR:R#="SXXXXXXXXXXXXXXXXXXXXXXXXXXXX":D
IM#(25),NC(30,5),NC#(30),C(60),F(8)
3 DATA3,100,82,70,64,67,68,69,99,32,1
00,111,121,98,248,247,227,160
5 PRINTCHR$(14):FORI=0T08:READC(I):NEX
T:FORI=0T08:READC(I):NEXT I:T6=160
10 N=8:FORI=1T025:N#(I)=LEFT$(R#,I):NE
XT:FORI=1T0N:F(I)=0:NEXT I:PRINT"Q"
20 F2=0:PRINTM$(1)TAB(15)"M E N U"
:PRINTM$(4)TAB(3)"33" -LINE
30 PRINTM$(6)TAB(3)"31" -INSERIMENTO
DATA"
40 PRINTM$(8)TAB(3)"32" -IDEFINIZIONE
E CLASSI DI"
50 PRINTTAB(7)"VALORI ARGOMENTALI"
60 PRINTM$(10)TAB(3)"33" -LAPPRESENTA
ZIONE SINTETICA"
70 PRINTM$(12)TAB(3)"34" -\DPA, MEDIA
NA, QUANTILI,"
75 PRINTTAB(7)"POPOLAZIONE TOTALE"
80 PRINTM$(14)TAB(3)"35" -LAPPRESENTA
ZIONE GRAFICA"

```

"Wei-ch'ii",  
un gioco cinese per PET

Segue figura 2.

```

90 PRINTW$(16)TAB(3)"CORREZIONE
DATI"
100 PRINTW$(18)TAB(3)"REGISTRAZI
ONE DATI SU NASTRO"
110 PRINTW$(20)TAB(3)"LETTURA DA
TI DA NASTRO"
120 GETR$:IFR$=""THEN120
130 R=VAL(R$):IFR<0ORR>THEN120
140 IF(R<0)ANDR<0)ANDF(1)=0THE
NGOSUB10000:GOTO20
150 IF(R<0)ANDR<0)ORF(1)<0)THEN170
160 GOSUB13000:IFF1=1THEN120
170 IFR=2ANDAB#<"B"THENGOSUB14000:GOT
O120
180 R=R+1:ONRGO SUB500,1000,2000,3000,4
000,5000,6000,7000,8000
190 IFF2=1THENEND
200 PRINT":GOTO20
500 F2=1:RETURN
1000 F(1)=1:BA#=""PRINTW$(1)TAB(10
)"INSERIMENTO DATI"
1010 PRINTW$(4)TAB(3)"ATI GIA'
DISTRIBUITI"
1015 PRINTTAB(7)"IN CLASSI ARGOMENTALI
"
1020 PRINTW$(6)TAB(3)"ATI NON O
STRIBUITI"
1025 PRINTTAB(7)"IN CLASSI ARGUMENTALI
"
1030 GETAB$:IFAB$=""THEN1030
1040 IFAB#<"A"ANDAB#<"B"THEN1030
1050 IFAB#<"B"THEN1500
1051 PRINTW$(1)TAB(10)"ATI DISTR
IBUITI"
1052 PRINTW$(4)TAB(3)"LASSI ARG
OMENTALI"
1053 PRINTTAB(7)"AD INTERVALLO NUMERIC
O"
1054 PRINTW$(6)TAB(3)"LASSI ARG
OMENTALI"
1055 PRINTTAB(7)"PARTICOLARI"
1056 GETBA$:IFBA$=""THEN1056
1057 IFBA#<"A"ANDBA#<"B"THEN1056
1058 IFBA#<"B"THENGOSUB2000:RETURN
1060 GOSUB15000:FORI=1TOC:PRINTW$(#F(
1)TAB(13)I"CLASSE"
1070 PRINTW$(4)TAB(3)"INTERVALLO DI CL
ASSE:"HC(I,0)"-HC(I,1)
1080 PRINTW$(6)TAB(3)"FREQUENZA INDIVI
DUI" I"CLASSE":PRINTTAB(3):INPUTHC(I,
2)
1085 IFNC(I,2)<ORNC(I,2)>99999THEN108
0
1087 IFNC(I,2)<INTNC(I,2)>THEN1080
1090 NEXT:GOSUB16000:RETURN
1500 PRINTW$(9)TAB(3)"NUMERO CAMPIONI
RACCOLTI"
1510 PRINTTAB(3)"(MASSIMO 60)":INPUTC
:IFC(10C)>60THEN1500
1515 FORJ=1TOC:J)=0:NEXT:FORI=1TOC:G
OSUB12500
1520 PRINTW$(12)TAB(3)"VALORE CAMPIONE
NR." I

```

Segue figura 2.

```

1525 PRINTTAB(3):INPUTC(I):IFC(I)<0TH
EN1520
1530 NEXT:GOSUB15000:GOSUB17000:GOSUB1
6000:RETURN
2000 GOSUB15000:GOSUB17000:GOSUB16000:
RETURN
3000 PRINT":F3=0:F(3)=1
3005 PRINTW$(1)TAB(5)"APPRESENTAZIO
NE SINTETICA"
3010 PRINTW$(4)TAB(3)"LINE":PRIN
TW$(6)TAB(3)"MEDIA"
3020 PRINTW$(8)TAB(3)"VALORE QUA
DRATICO MEDIO"
3030 PRINTW$(10)TAB(3)"VARIANZA"
3040 PRINTW$(12)TAB(3)"CARTO QU
ADRATICO MEDIO"
3050 PRINTW$(14)TAB(3)"INDICE DI
SKEWNESS"
3060 PRINTW$(16)TAB(3)"EFFICIE
NTE GAMMA"
3070 PRINTW$(18)TAB(3)"MOMENTO K
-ESIMO RISPETTO"
3080 PRINTTAB(7)"AL POLO THETA"
3090 GETRS$:IFRS$=""THEN3090
3100 RS=VAL(R$):IFRS<0ORRS>THEN3090
3110 RS=RS+1:ONRSGO SUB3130,3150,3200,3
300,3400,3500,3600,3700
3120 IFF3=1THENRETURN
3123 IFF4=1THENF4=0:PRINT":GOTO3005
3125 GOSUB11000:GOTO3090
3130 F3=1:RETURN
3150 GOSUB18000:PRINTW$(22)TAB(3)"MED
IA:V:RETURN
3200 VOM=0:FORI=1TOC:VOM=VOM+(NC(I,5)
(2)*NC(I,3):NEXT
3210 PRINTW$(22)TAB(3)"VALORE QUADRAT
ICO":PRINTTAB(3)"MEDIA:V:RETURN
3300 GOSUB19000:PRINTW$(22)TAB(3)"VAR
IANZA:V:RETURN
3400 GOSUB20000:PRINTW$(22)TAB(3)"COR
RTO QUADRATICO MEDIO:V:RETURN
3500 IFBA#<"B"THENGOSUB29000:RETURN
3505 SK=0:GOSUB18000:GOSUB21000:IFZM=1
THENS#<"ZEROMODALE":GOSUB22000:RETURN
3510 IFPM=1THENS#<"PLURIMODALE":GOSUB2
2000:RETURN
3520 GOSUB20000:SK=(M-MODA)/SOM
3530 PRINTW$(22)TAB(3)"INDICE DI SKEW
NESS:V:SK:RETURN
3600 IFBA#<"B"THENGOSUB29000:RETURN
3605 CG=0:GOSUB18000:GOSUB20000:FORI=1
TOC:CG=CG+(NC(I,5)-M(1,3))*NC(I,3):NEXT
3607 CG=CG/(SOM(3)):IFAB$(CG)<9E-9THENC
0=0
3610 PRINTW$(22)TAB(3)"EFFICIENTE G
AMMA:V:CG:RETURN
3700 PRINT":F4=0
3705 PRINTW$(1)TAB(3)"MOMENTO K-ESIM
O AL POLO THETA"
3707 PRINTW$(3)TAB(3)"SPONENTE K=":I
NPUT:PRINTW$(5)TAB(3)"THETA"
3710 PRINTW$(7)TAB(3)"LINE":PRIN
TW$(9)TAB(3)"MEDIA"
3720 PRINTW$(11)TAB(3)"MODA":PRI
NTW$(13)TAB(3)"VARIANZA":PRINTW$(15)TAB(3)"QUANTI
LE"
3740 PRINTW$(17)TAB(3)"QUANTILE"
3750 PRINTW$(19)TAB(3)"HETA GEN
ERICO"

```

**"Wei-ch'i",  
un gioco cinese per PET**

Segue figura 2.

```

3760 GETKT#:IFKT#="" THEN3760
3770 KT=VAL(KT#):IFKT<0ORKT>6THEN3760
3780 KT=KT+1:ONKGOOSUB3810,3820,3850,3
890,3920,3950,3970
3790 IFF4=1THENPRINT"Q":RETURN
3800 GO SUB11000:GOTO3760
3810 F4=1:RETURN
3820 TH=MODA:GOSUB23000:PRINTW*(22)TAB
(3)"MOMENTO DI INDICE:MK"
3830 PRINTTAB(3)"SAVENTE PER POLO":PR
INTTAB(3)"LA MEDIA:MK:RETURN
3850 IFBA#="B"THENGO SUB29000:RETURN
3855 GO SUB21000:IFZM=1THENS#="ZEROMODA
LE":GOSUB22000:RETURN
3855 IFM=1THENS#="PLURIMODALE":GOSUB2
2000:RETURN
3860 TH=MODA:GOSUB23000:PRINTW*(22)TAB
(3)"MOMENTO DI INDICE:MK"
3870 PRINTTAB(3)"SAVENTE PER POLO":PR
INTTAB(3)"LA MODA:MK:RETURN
3890 IFBA#="B"THENGO SUB29000:RETURN
3905 GO SUB24000:TH=ME:GOSUB23000:PRINT
W*(22)TAB(3)"MOMENTO DI INDICE:MK"
3900 PRINTTAB(3)"SAVENTE PER POLO":PR
INTTAB(3)"LA MEDIANA:MK:RETURN
3920 IFBA#="B"THENGO SUB29000:RETURN
3925 GO SUB25000:TH=Q1:GOSUB23000:PRINT
W*(22)TAB(3)"MOMENTO DI INDICE:MK"
3930 PRINTTAB(3)"SAVENTE PER POLO IL":
:PRINTTAB(3)"QUANTILE:MK:RETURN
3950 IFBA#="B"THENGO SUB29000:RETURN
3955 GO SUB26000:TH=Q3:GOSUB23000:PRINT
W*(22)TAB(3)"MOMENTO DI INDICE:MK"
3960 PRINTTAB(3)"SAVENTE PER POLO IL":
:PRINTTAB(3)"QUANTILE:MK:RETURN
3970 PRINTW*(21)TAB(3)"I HETA =":INPUT
TH:GOSUB23000
3980 PRINTW*(22)TAB(3)"MOMENTO DI IN
DICE:MK"
3990 PRINTTAB(3)"SAVENTE POLO THETA:MK"
TH:PRINTTAB(3)"(K, THETA):MK:RETURN
4000 PRINT"Q":F5=0:F(4)=1
4005 PRINTW*(1)TAB(7)"MODA, MEDIANA,
QUANTILI, "
4007 PRINTTAB(7)"POPOLAZIONE TOTA
LE "
4010 PRINTW*(4)TAB(3)"LINE _LINE"
PRINTW*(6)TAB(3)"MODA _MODA"
4020 PRINTW*(8)TAB(3)"MEDIA _MEDIANA"
4030 PRINTW*(10)TAB(3)"QUANT _QUANTI
LE"
4040 PRINTW*(12)TAB(3)"QUANT _QUAN
TILE"
4045 PRINTW*(14)TAB(3)"POPOLAZIO
NE TOTALE"
4050 GETM#:IFM#="" THEN4050
4060 M=VAL(M#):IFM<0ORM#>5THEN4050
4070 M=M+1:ONMGOOSUB4095,4100,4200,4
300,4400,4500
4080 IFF5=1THENRETURN
4090 GO SUB11000:GOTO4050
4095 F5=1:RETURN
4100 IFBA#="B"THENGO SUB29000:RETURN

```

Segue figura 2.

```

4105 GO SUB21000:IFZM=1THENS#="ZEROMODA
LE":GOSUB22000:RETURN
4110 IFM=1THENS#="PLURIMODALE":GOSUB2
2000:RETURN
4120 PRINTW*(22)TAB(3)"MODA:MK:MODA:RE
TURN
4200 IFBA#="B"THENGO SUB29000:RETURN
4205 GO SUB24000:PRINTW*(22)TAB(3)"MED
IANA:MK:RETURN
4300 IFBA#="B"THENGO SUB29000:RETURN
4305 GO SUB25000:PRINTW*(22)TAB(3)"QUA
NTILE:MK:RETURN
4400 IFBA#="B"THENGO SUB29000:RETURN
4405 GO SUB26000:PRINTW*(22)TAB(3)"QUA
NTILE:MK:RETURN
4500 PRINTW*(22)TAB(3)"POPOLAZIONE TOTA
LE:MK:RETURN
5000 PRINT"Q":F9=0:F(5)=1:CA=1
5005 PRINTW*(1)TAB(6)"APPRESENTAZIO
NE GRAFICA "
5010 PRINTW*(4)TAB(3)"LINE _LINE"
5020 PRINTW*(6)TAB(3)"STOGRAMMA
DELLA FREQUENZA"
5025 PRINTTAB(7)"ASSOLUTA"
5035 PRINTW*(8)TAB(3)"STOGRAMMA
DELLA FREQUENZA"
5035 PRINTTAB(7)"RELATIVA"
5040 PRINTW*(10)TAB(3)"STOGRAMM
A DELLA FREQUENZA"
5045 PRINTTAB(7)"CUMULATA"
5050 GETRG#:IFRG#="" THEN5050
5060 RG=VAL(RG#):IFRG<0ORRG>3THEN5050
5065 IFBA#="B"ANDRG=3THENGO SUB34000:GO
TO5000
5070 RF=RG+1:ONRGOOSUB5095,5100,5200,5
300
5080 IFF9=1THENRETURN
5090 GOTO5005
5095 F9=1:RETURN
5100 GO SUB21000:XM=X:T#=# FREQUENZA A
SSOLUTA " :GOSUB31000
5110 IFBA#="B"THENXM=(INT(XM)+1)
5120 DU=XM/T6:GOSUB37000:W=2:GOSUB3600
0
5199 GO SUB32000:POKE59468,14:RETURN
5200 T#=# _REQUENZA RELATIVA "
5205 GO SUB31000:DU=1/T6:GOSUB33000:W=3
:IFBA#="B"THENW=1
5299 GO SUB35000:GOSUB32000:RETURN
5300 T#=# _REQUENZA CUMULATA "
5305 GO SUB31000:DU=1/T6:GOSUB33000:W=4
:GOSUB35000
5399 GO SUB32000:RETURN
6000 PRINT"Q":F6=0:F(6)=1
6005 PRINTW*(1)TAB(10)"CORREZIONE, DA
TI "
6010 IFBA#="B"THEN6500
6020 IFBA#="B"THEN6250
6030 FORI=1TOUN:PRINT"Q" W*(1)TAB(13)"I
"
6040 PRINTW*(4)TAB(3)"INTERVALLO DI CL
ASSE:"NC(I,0)="-NC(I,1)
6050 PRINTW*(6)TAB(3)"_REQUENZA INDIVI

```

"Wei-ch'i",  
un gioco cinese per PET

Segue figura 2.

```

DUI":PRINTTAB(3)I"" CLASSE"HC(I,2)
6060 PRINTM$(9)TAB(3)"ESIDERI CAMBIA
RE LA FREQUENZA"
6070 GETCF$:IFCF$=""THEN6070
6080 IFCF$<>"S"ANDCF$<>"N"THEN6070
6090 IFCF$="N"THEN6110
6100 PRINTM$(12)TAB(3)"UOVA FREQUENZA
INDIVIDUI"
6105 PRINTTAB(3)I"" CLASSE":INPUTNC(I
,2)
6110 NEXT:GOSUB16000:RETURN
6250 FORI=1T0NC:PRINT"U"$(1)TAB(13)I"
CLASSE "
6260 PRINTM$(4)TAB(3)"ARATTERISTICA D
I CLASSE":PRINTTAB(3)HC(I,2)
6270 PRINTM$(6)TAB(3)"XALORE ASSOCIATO
ALLA":PRINTTAB(3)I"" CLASSE"HC(I,2)
6280 PRINTM$(9)TAB(3)"ESIDERI CAMBIA
RE IL VALORE"
6290 GETCF$:IFCF$=""THEN6290
6300 IFCF$<>"S"ANDCF$<>"N"THEN6290
6310 IFCF$="N"THEN6330
6320 PRINTM$(12)TAB(3)"UOVO VALORE AS
SOCIATO ALLA"
6325 PRINTTAB(3)I"" CLASSE":INPUTNC(I
,2)
6330 NEXT:GOSUB16000:RETURN
6500 FORI=1T0C:PRINT"U"$(1)TAB(13)"A
MPIONE NR."I
6510 PRINTM$(4)TAB(3)"XALORE CAMPIONE
NR."I:PRINTTAB(3)C(I)
6520 PRINTM$(7)TAB(3)"ESIDERI CAMBIA
RE IL VALORE"
6530 GETCF$:IFCF$=""THEN6530
6540 IFCF$<>"S"ANDCF$<>"N"THEN6530
6550 IFCF$="N"THEN6570
6560 PRINTM$(10)TAB(3)"UOVO VALORE DE
L CAMPIONE"
6565 PRINTTAB(3)"NR."I":INPUTC(I)
6570 NEXT:FORI=1T0NC:FORJ=2T04:HC(I,J)
=0:NEXTJ,I:GOSUB17000:GOSUB16000:RETURN
7000 PRINT"U"$(1)TAB(4)"EGISTRAZIO
NE DATI SU NASTRO "
7010 PRINTM$(4)TAB(3)"OME DISTINTIVO
DEL FILE"
7020 PRINTTAB(3)"SOLO 15 CARATTERI SI
GNIFICATIVI":PRINTTAB(3):INPUTNF$
7040 PRINTM$(7)TAB(3)"OCURATEVI UNA
CASSETTA E RIAV-"PRINTTAB(3)"VOLGETELA
"
7050 PRINTTAB(3)"FRONTI":INPUTZ$:IFLE
FT$(Z$,1)<>"S"THEN7040
7055 IFBA$=""THENBA$="C"
7057 IFCV$=""THENCV$="Z"
7060 OPEN1,1,2,NF$:PRINT#1,NC:PRINT#1,
C:INPUT#1,F1:PRINT#1,AB$:PRINT#1,BH$
7070 FORI=1T08:PRINT#1,F(I):NEXT:FORI=
1T0NC:FORJ=0T05:PRINT#1,HC(I,J):NEXTJ
7080 NEXTI:FORI=1T0C:PRINT#1,C(I):NEXT
7090 PRINT#1,CV$:IFBA$="B"THENFORI=1T0
NC:PRINT#1,NC$(I):NEXT
7095 IFBA$="C"THENBA$=""
7097 IFCV$="Z"THENCV$=""

```

Segue figura 2.

```

7100 CLOSE1:RETURN
8000 PRINT"U"$(1)TAB(7)"LETTURA DAT
I DA NASTRO "
8010 PRINTM$(4)TAB(3)"OME DISTINTIVO
DEL FILE":PRINTTAB(3):INPUTNF$
8020 PRINTM$(7)TAB(3)"LIVVOLGETE LA C
ASSETTA CONTE"
8030 PRINTTAB(3)"ENTE IL FILE DATI."
8040 PRINTTAB(3)"FRONTI":INPUTZ$:IFLE
FT$(Z$,1)<>"S"THEN8020
8060 OPEN1,1,0,NF$:INPUT#1,NC:INPUT#1,
C:INPUT#1,FT:INPUT#1,AB$:INPUT#1,BA$
8070 FORI=1T08:INPUT#1,F(I):NEXT:FORI=
1T0NC:FORJ=0T05:INPUT#1,HC(I,J):NEXTJ
8080 NEXTI:FORI=1T0C:INPUT#1,C(I):NEXT
8090 INPUT#1,CV$:IFBA$="B"THENFORI=1T0
NC:PRINT#1,NC$(I):NEXT
8095 IFBA$="C"THENBA$=""
8097 IFCV$="Z"THENCV$=""
8100 CLOSE1:RETURN
10000 PRINTM$(22)TAB(3)"ON SONO DISP
ONIBILI DATI"
10010 PRINTTAB(3)"SU CUI OPERARE":G
OSUB11000:RETURN
11000 FORI=1T05000:NEXT:GOSUB12000:RET
URN
12000 FORA=33568T033768:POKEA,32:NEXT:
RETURN
12500 FORA=33208T033408:POKEA,32:NEXT:
RETURN
13000 F1=0:PRINTM$(22)TAB(3)"UOI DIS
TRUGGERE I DATI"
13010 PRINTTAB(3)"ESISTENTI":INPUTR
1$:R1$=LEFT$(R1$,1)
13020 IFR1$<>"S"ANDR1$<>"N"THEN13000
13030 IFR1$="S"THENGOSUB12000:RETURN
13040 F1=1:GOSUB12000:RETURN
14000 PRINTM$(22)TAB(3)"ON SONO DISP
ONIBILI DATI"
14010 PRINTTAB(3)"ISAGGREGATI":GOS
UB11000:RETURN
15000 PRINT"U"$(1)TAB(4)"EFINIZION
E CLASSI ARGOMENTALI "
15005 PRINTM$(4)TAB(3)"UMERO CLASSI A
RGOMENTALI"
15007 PRINTTAB(3)"DESIDERATE (MASSIMO
30)":INPUTNC
15008 IFNC<20RNC>30THEN15005
15009 FORJ=1T0NC:FORK=1T05:HC(J,K)=0:N
EXTK,J
15010 PRINTM$(7)TAB(3)"LASSI -LASSI AD
AMPIEZZA COSTANTE"
15020 PRINTM$(9)TAB(3)"LASSI -LASSI AD
AMPIEZZA VARIABILE"
15030 GETCV$:IFCV$=""THEN15030
15040 IFCV$<>"C"ANDCV$<>"V"THEN15030
15050 PRINTM$(12)TAB(3)"LIMITE INFERIO
RE DELLA"
15060 PRINTTAB(3)"I" CLASSE":INPUTNC(
1,0)
15065 IFCV$="V"THEN15500
15070 PRINTM$(14)TAB(3)"MPIEZZA DELLE
CLASSI":PRINTTAB(3):INPUTAC

```



"Wei-ch'i",  
un gioco cinese per PET

Segue figura 2.

```

28030 NC(I)=LEFT$(NC*(I)+
"
28040 PRINTW$(14)TAB(3)"XALORE ASSOCIA
TO ALLA"
28050 PRINTTAB(3)I" CLASSE";INPUTNC(I
,2):IFNC(I,2)>999999ORNC(I,2)<0THEN2804
0
28055 NC(I,5)=1;GOSUB12500;NEXT
28060 GOSUB16000;RETURN
29000 PRINTW$(22)TAB(3)"L'ORGANIZZAZIO
NE DELLA VARIABILE"
29010 PRINTTAB(3)"STATISTICA RENDE IL
CALCOLO DEL-"
29020 PRINTTAB(3)"L'INDICE PRIVA DI SI
GNIFICATO.";RETURN
30000 IFB#<>"B"THENRETURN
30010 FORI=1TOUNC(I,1)=NC(I,3):NC(I,
3)=1/NC(NC(I,5)=NC(I,2):NEXT;RETURN
31000 PRINT"Q";PRINTCHR$(142):POKE5946
8,14
31005 AC=INT(30/NC):FORI=1TOINT(NC/2)+
1:FORJ=1TOAC
31007 IFNC=INT(NC/2)*2ANDI>NC/2THEN310
20
31009 POKE(33654+J+AC*(I-1)*2),99
31010 POKE(33574+J+(I-1)*AC*2),99:IFNC
<>INT(NC/2)*2ANDI>NC/2THEN31020
31015 POKE(33574+J+AC*(I-1)*2+AC),227
31017 POKE(33654+J+AC*(I-1)*2+AC),99
31020 NEXT;NEXT;FORI=1TO2:POKE(33654-
I*40),160;NEXT
31040 POKE33654,99:FORI=1TO2:POKE(3365
4-I*40+AC*NC+1),101;NEXT
31499 GOSUB38000;RETURN
32000 GOSUB16700;PRINT"Q";PRINTCHR$(14
):RETURN
33000 FORI=1TO10;PRINTW$(I*2+1)"0. 00"
33005 POKE(32767+(I)*#0+6),A(8):POKE(3
2767+(I)*#0+7),A(8)+128
33010 POKE(32767+(I)*#0+3),(48+(I-1))
;NEXT;PRINTW$(1)"1.000"
33999 RETURN
34000 PRINTW$(22)TAB(3)"L'ORGANIZZAZIO
NE DELLA VARIABILE"
34010 PRINTTAB(3)"STATISTICA RENDE LA
RAPPRESENTA-"
34020 PRINTTAB(3)"ZIONE PRIVA DI SIGNI
FICATO.";GOSUB11000;RETURN
35000 FORI=1TOUNC(I)=INT(20*NC(I,W)):08
=INT(UNC(I,W)/DU)-01#0):IF08<0THEN08=0
35010 IF08>8THEN08=8
35015 IF01=0THEN35025
35020 FORJ=1TO01:FORK=1TOAC:POKE(33574
-J*40+K+AC*(I-1)),160;NEXT;NEXT
35025 IF08>8THEN35040
35030 FORK=1TOAC:POKE(33574-(01+1)*40+
K+AC*(I-1)),B(08):NEXT;NEXT;RETURN
35040 NEXT
35050 RETURN
36000 FORI=1TOUNC(I)=INT(20*(NC(I,W)/WM
)):08=INT(UNC(I,W)/DU)-01#0)
36015 IF01=0THEN36025
36020 FORJ=1TO01:FORK=1TOAC:POKE(33574

```

Segue figura 2.

```

-J*40+K+AC*(I-1)),160;NEXT;NEXT
36025 IF08<0THEN35040
36030 FORK=1TOAC:POKE(33574-(01+1)*40+
K+AC*(I-1)),B(08):NEXT;NEXT;RETURN
36040 NEXT
36050 RETURN
37000 POKE59468,12;CF=0
37005 POKE(33574),A(8)+128
37010 IFXND=10+CFANDXND<10+CF+1)THEN37
030
37020 CF=CF+1;GOTO37010
37030 SP=(10+CF):FORJ0=0TOXNSTEPS:J=J
0+1000;X0=XM*1000;D0=DU*1000
37033 QI=INT(20*(J/X0)):08=INT((J/D0)-
QI#0+CA)
37035 IF08>8THEN08=8
37037 TB=(20-QI):IF(20-QI)=0THENTB=1
37040 J$=STR$(INT(J0)):J#=RIGHT$(
" "+J$,5):PRINTW$(TB)J$
37045 IF(33573-(01+1)*40)<32758THENQ1
=01-1;08=05
37050 POKE(33573-(01+1)*40),A(08)
37060 POKE(33574-(01+1)*40),A(08)+128
37070 NEXT
37080 FORJ0=(SP/2)TOXNSTEPS:J=J0+1000
;X0=XM*1000;D0=DU*1000
37083 QI=INT(20*(J/X0)):08=INT((J/D0)-
QI#0+CA)
37085 IF08>8THEN08=8
37087 IFXND(10+(CF+1))/2THEN37100
37088 TB=(20-QI):IF(20-QI)=0THENTB=1
37090 J$=STR$(INT(J0*10)/10):J#=RIGHT$(
" "+J$,6):PRINTW$(TB)J$
37100 POKE(33574-(01+1)*40),A(08)+128
37110 NEXT;RETURN
38000 IFAC=1THEN38500
38010 FORI=1TOUNC(I$=STR$(I):I#=RIGHT$(
" "+I$,LEN(I$)-1)
38020 I#=RIGHT$( " "+I$,AC):P
RINTW$(22)TAB(7+AC*(I-1))I$;NEXT
38030 FORI=1TOINT(NC/2)+1:FORJ=1TOAC:C
H=PEEK(33614+J+(I-1)*AC*2):CH=CH+128
38035 IFNC=INT(NC/2)*2ANDI>NC/2THEN380
50
38040 POKE(33614+J+(I-1)*AC*2),CH
38050 NEXT;NEXT;RETURN
38500 FORI=1TOUNC(I)=INT(I/10):C2=I-01*
10
38510 C1$=STR$(C1):C1#=RIGHT$( " "+C1$
,LEN(C1$)-1)
38520 C1#=RIGHT$( " "+C1$,AC)
:IFC1=0THENC1$=""
38525 PRINTW$(22)TAB(7+AC*(I-1))C1$
38530 C2$=STR$(C2):C2#=RIGHT$( " "+C2$
,LEN(C2$)-1)
38540 C2#=RIGHT$( " "+C2$,AC)
;PRINTW$(23)TAB(7+AC*(I-1))C2$;NEXT
38550 FORI=1TOINT(NC/2)+1:FORK=0TO1:FO
R=L=1TOAC
38560 CH=PEEK(33614+K*40+J+(I-1)*AC*2)
:CH=CH+128
38570 IFNC=INT(NC/2)*2ANDI>NC/2THEN385
90

```



# "Wei-ch'i", un gioco cinese per PET

### Segue figura 2.

```

38500 POKE(33614+K*40+J+(I-1)*AC*2),CH
38590 NEXT: NEXT: NEXT: POKE(33654,160:POKE
E(33654+AC*NC+1),101
38600 FOR I=33694 TO (33694+AC*NC):POKE I,
99:NEXT: RETURN
38999 RETURN
39000 CF=0
39010 IF XM=10↑CF AND XM<10↑(CF+1) THEN 39
030
39020 CF=CF+1:GOTO 39010
39030 CF=5-CF: RETURN
39500 N0#=NC*(I+J*10):N2=INT(NC(I+J*10
,2)*(10↑CF))/10↑CF)
39999 N2#=-STR$(N2):N2#=-RIGHT$("
"+N2#,7): RETURN

```

### Lista simboli grafici

- 0 : 1 HOME =CHR\$(19)
- 24 CRSR↑ =CHR\$(17)
- 10 : 1 SHIFT HOME =CHR\$(147)
- 20 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 30 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 40 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 60 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 70 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 80 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 90 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 100 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 110 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 200 : 1 SHIFT HOME =CHR\$(147)
- 1000 : 1 SHIFT HOME =CHR\$(147)
- 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1010 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1020 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)

### Segue figura 2.

- 1051 : 1 SHIFT HOME =CHR\$(147)
- 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1052 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1054 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1050 : 1 SHIFT HOME =CHR\$(147)
- 3000 : 1 SHIFT HOME =CHR\$(147)
- 3005 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3010 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3020 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3030 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3040 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3050 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3060 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3070 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3123 : 1 SHIFT HOME =CHR\$(147)
- 3150 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3210 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3300 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3400 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3530 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3610 : 1 REVERSE =CHR\$(18)
- 1 SHIFT REVERSE =CHR\$(146)
- 3700 : 1 SHIFT HOME =CHR\$(147)



# "Wei-ch'i", un gioco cinese per PET

### Seguito figura 2.

```

3705 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3707 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3710 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3720 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3730 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3740 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3750 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3790 : 1 SHIFT HOME =CHR$(147)

3820 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3830 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3860 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3870 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3895 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3900 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3925 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3930 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3955 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

```

### Seguito figura 2.

```

3960 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3990 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

3990 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4000 : 1 SHIFT HOME =CHR$(147)

4005 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4007 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4020 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4030 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4040 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4045 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4120 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4205 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4305 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4405 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

4500 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

5000 : 1 SHIFT HOME =CHR$(147)

5005 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

5010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

5020 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

5030 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

5040 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

```

**"Wei-ch'i",  
un gioco cinese per PET**

*Seguito figura 2.*

```

5100 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

5200 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

5300 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

6000 : 1 SHIFT HOME =CHR$(147)

6005 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

6030 : 1 SHIFT HOME =CHR$(147)

6060 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

6250 : 1 SHIFT HOME =CHR$(147)

6280 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

6500 : 1 SHIFT HOME =CHR$(147)

6520 : 1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

7000 : 1 SHIFT HOME =CHR$(147)
      1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

8000 : 1 SHIFT HOME =CHR$(147)
      1 REVERSE =CHR$(18)
      1 SHIFT REVERSE =CHR$(146)

10000 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

10010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

13000 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

13010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

14000 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

14010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

15000 : 1 SHIFT HOME =CHR$(147)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

15010 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

```

*Seguito figura 2.*

```

15020 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

16505 : 1 SHIFT HOME =CHR$(147)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

16510 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

16520 : 1 SHIFT HOME =CHR$(147)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

16530 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

16700 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)
        1 SHIFT CSRRT =CHR$(145)

17000 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

17002 : 1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

28000 : 1 SHIFT HOME =CHR$(147)
        1 REVERSE =CHR$(18)
        1 SHIFT REVERSE =CHR$(146)

31000 : 1 SHIFT HOME =CHR$(147)

32000 : 1 SHIFT HOME =CHR$(147)

```

# VIDEO Giochi

**LA PRIMA E UNICA  
RIVISTA DI VIDEOGAMES  
COMPUTER  
GIOCHI ELETTRONICI**



Una pubblicazione  
del Gruppo Editoriale Jackson



# Servizio programmi

Per alcuni dei programmi pubblicati, *Personal Software* mette a disposizione dischi e nastri già registrati, realizzati in collaborazione con l'autore. Potete ottenerli in contrassegno, pagando direttamente al postino la cifra indicata, spedendo il tagliando pubblicato in fondo alla pagina.

N.	Sistema	Programmi	Supporto	pubblicato in <i>Personal Software</i> n.	Prezzo
1	Apple II+	La carta del cielo Collisione	floppy 5" DOS 3.3	3 pag. 83 3 pag. 93	30.000
2	TRS-80 mod. I	Backgammon	floppy 5" DOS 2.3	3 pag. 89	25.000
3	PET/CBM 3032/4032	Editor/Assembler in Basic	floppy 5" 3032/4032+3040/4040	2 pag. 33	40.000
4	Apple II+	Interi in precisione multipla Grafica 3D	floppy 5" DOS 3.3	4 pag. 17 4 pag. 47	40.000
5	PET/CBM 3032/4032	Gioco del calcio	floppy 5" 3032/4032+3040/4040	4 pag. 67	25.000
6	Apple II +	Pretty Printer Shape Table	floppy 5" DOS 3.3	5 pag. 27 5 pag. 58	30.000
7	Apple II +	Data base modulare	floppy 5" DOS 3.3	7 pag. 47	25.000
8	PET/CBM 3032/4032	Wei-ch'i	cassetta	12/13 pag. 34	20.000

Spedire in busta  
chiusa a

PERSONAL SOFTWARE  
Servizio Programmi  
Via Rosellini 12  
20124 Milano

Inviatemi i seguenti dischi di *Personal Software*

n. \_\_\_\_\_

per un totale di lire \_\_\_\_\_ + L. 2.000 come contributo fisso  
spese di spedizione che pagherò al postino alla consegna del pacco.

Cognome e nome \_\_\_\_\_

Indirizzo \_\_\_\_\_

Cap., Località \_\_\_\_\_

Firma \_\_\_\_\_

# un nuovo corso per imparare a dialogare con il personal computer



## Il personal computer: un protagonista

Il computer è figlio dell'informatica, la scienza degli anni '80, che sta rivoluzionando il mondo della produzione e, in un futuro non molto remoto, trasformerà radicalmente la qualità della nostra vita.

I computers sono ormai pronti a lavorare per noi: ora siamo noi che dobbiamo imparare a comunicare con loro, per metterli in grado, con le nostre istruzioni, di fornirci il maggior numero di prestazioni e al più elevato livello.

È in quest'ottica che INFOR ha messo a punto il suo corso di informatica di base programmato per l'insegnamento a distanza, che rappresenta lo strumento più perfezionato oggi reperibile per chi vuole trattare da pari a pari con il proprio elaboratore.

## Un corso per tutti

Questo corso teorico-pratico è indispensabile per chiunque, già inserito nel mondo o nel mercato del lavoro, desideri accostarsi all'informatica per migliorarne le proprie capacità produttive. E anche per i giovani che vogliono inserirsi in questo nuovo mondo.

## La gestione del-computer; il linguaggio BASIC

Il corso è facilmente comprensibile a chi si avvicina per la prima volta all'informatica: ne insegna l'abc con un linguaggio semplice e piano, e mettendo fin dalla prima lezione l'utente a contatto diretto con il personal, in poche settimane e gli insegna, comodamente a casa, a farlo funzionare a programmarlo in BASIC. A fine corso INFOR rilascia un attestato a conferma della preparazione raggiunta.

Altre 7 proposte INFOR: Giornalista, Fotografo, Interprete, Grafico pubblicitario, Tecnico pubblicitario, Programmista radio-TV, Audiovisivi.



ISTITUTO SUPERIORE PER LA  
COMUNICAZIONE E L'INFORMAZIONE

**INFOR** Via G.V. Englen, 25/T 00163 Roma

**INFORMAZIONI URGENTI:**  
TEL. 06 62.30.341

Desidero ricevere informazioni sul vostro corso \_\_\_\_\_

Cognome \_\_\_\_\_ Nome \_\_\_\_\_ Età \_\_\_\_\_

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

Città \_\_\_\_\_ C.A.P. \_\_\_\_\_

Motivo della richiesta  studio  lavoro  hobby

**W 5 2 0**

# Linguaggio macchina per ZX81

## Alla scoperta dei principi dell'Assembler attraverso il BASIC

di Bruno Del Medico

**Q**uesto articolo e due altri successivi sono dedicati a tutti coloro che vogliono capire dal principio che cosa sia il linguaggio macchina.

Agli interessati si presenterà un abbozzo di programma Assembler funzionante e modificabile per integrazioni e miglioramenti. Siamo convinti che molti lettori dopo aver seguito questo primo articolo, saranno entusiasti di partecipare alla costruzione dell'Assembler ZX81 PS.

In questo articolo si trattano le istruzioni più semplici e i metodi di memorizzazione fornendo 4 listati di giochi di movimento.

### Analogie con il BASIC

Cominciamo con l'osservare alcune analogie formali tra il BASIC e il LM.

Le operazioni svolte sono differenti nella sostanza ma il confronto è il metodo più diretto per capire.

LD A, 15	LET A = 21
ADD A, 15	LET A = A + 21
LD C, A	LET C = A
LD B, 0	LET B = 0
RET	RETURN

LD A, 15 equivale a LET A = 21 perchè 15 è un numero esadecimale. (Vedi il riquadro: conversione decimale-esadecimale).

LD è una contrazione di LOAD, che significa *carica, metti in*.

ADD significa *aggiungi o somma*. La variabile A della routine BASIC viene immagazzinata nella parte di memoria RAM riservata alle variabili del programma. Se quest'area iniziasse all'indirizzo 17317, questa occupa i byte dal 17317 al 17322.

La variabile A della routine LM della memoria è di un particolare byte, che si chiama *registro* e si trova nella CPU dello Z80.

Il linguaggio BASIC permette l'uso di una notevole ed espandibile quantità di variabili numeriche.

Con il LM invece si dispone di un numero molto limitato di variabili o registri, tutti già predefiniti nella CPU.

Nella routine BASIC mancano i numeri di linea.

Per caricarla è necessario aggiungere:

```
10 LET A = 21
20 LET A = A + 21
30 LET C = A
40 LET B = 0
50 RETURN
```

Se si introduce la linea 10, le istruzioni vengono immagazzinate nella memoria a partire dall'indirizzo 16509, in questo modo:

```
16509 e 16510 contengono il numero di linea
16511 e 16512 contengono la lunghezza in byte della linea
16513 contiene LET
16514 contiene A
16515 contiene =
16516 fino a 16523 contengono il numero 21
16524 contiene NEWLINE
```

Seguono 793 byte riservati alla memoria di schermo (oppure 25 se non è usata l'espansione) e inizia poi l'area delle variabili del programma; qui i byte dal 17317 al 17322 contengono la variabile A. (Vedi la tabella 1).

Il programma BASIC immagazzinato negli indirizzi 16509/16524 definisce la variabile A (immagazzinata negli indirizzi 17317/17322) che vale 21 decimale, e poi ecc. ecc.

Analogamente dobbiamo sistemare la routine LM in memoria. La routine attribuisce a C il registro A della CPU il valore 15 esadecimale, ecc. ecc..

Le linee della routine BASIC possono essere numerate a piacere, ma in ogni caso vengono immagazzinate a partire dall'indirizzo 16509.

Con il LM la routine può venire immagazzinata a partire da qualsiasi indirizzo della memoria RAM, ma non si possono assegnare numeri di linea a piacere; i numeri di linea di ogni istruzione sono l'indirizzo di memoria in cui essa viene memorizzata.

In pratica, non sarà necessario indicare l'indirizzo per ogni singola istruzione; basta indicare l'indirizzo di partenza, e le istruzioni si intendono immagazzinate in sequenza:

```
17000 LD A, 15
      ADD A, 15
      LD C, A
      LD B, 0
      RET
```

Una prima conseguenza è che non sono concessi ripensamenti o inserimenti tardivi di linee.

Se vogliamo ottenere un output dalla routine BASIC, dobbiamo aggiungere una di queste due linee: 45 PRINT A oppure 45 PRINT B

Se invece vogliamo ottenere un output dalla routine LM, dobbiamo

## Linguaggio macchina per ZX81

scrivere:

```
1 PRINT USR 17000
```

che significa:

esegui la routine LM che comincia all'indirizzo 17000, e scrivi sullo schermo il valore dei registri BC al termine della routine.

Ecco quindi alcune caratteristiche del LM: sullo ZX81

- a) la routine LM deve sempre essere compresa o lanciata in un programma BASIC, poichè necessita dell'istruzione USR per essere eseguita;
- b) qualunque sia il risultato delle elaborazioni svolte dalla routine LM, deve essere sempre immagazzinato nei registri B e C per diventare accessibile alla istruzione USR. L'istruzione USR non fa altro che leggere il contenuto dei registri B e C quando la routine è terminata. Ora è chiaro il perchè delle istruzioni LD C, A e LD B, 0;
- c) i registri B e C forniscono solo un output numerico.

Esiste un solo metodo per scrivere qualcosa nella locazione di memoria 17000. Usare l'istruzione:

```
POKE 17000, qualcosa.
```

Solo che la parola *qualcosa* deve essere un numero, come richiesto dalla sintassi della istruzione POKE, mentre invece si vuole scrivere:

```
LD A, 15
```

In effetti questa è una istruzione in linguaggio Assembly e non può essere caricata nella memoria dello ZX81, prima bisogna tradurre le istruzioni Assembly nel loro corrispondente *codice oggetto esadecimale*.

Osservando la tabella 2 si nota che alcune istruzioni (per esempio LD A, 15) vengono immagazzinate in due byte diversi, altre in uno solo.

Anche nel BASIC dati ed istruzioni richiedono quantità diverse di byte per essere immagazzinati, ed è compito del sistema saperli leggere correttamente.

Qualcuno obietterà che neppure il codice oggetto può essere caricato con le POKE, perchè è esadecimale ma può essere caricato facilmente il suo equivalente decimale.

Quindi il metodo più semplice per caricare in memoria la routine LM è di battere una serie di POKE associate agli equivalenti decimali (terza colonna):

```
POKE 17000, 62
```

```
POKE 17001, 21
```

```
POKE 17002, 198
```

... e così via.

Successivamente con delle istruzioni PEEK possiamo verificare l'avvenuto buon caricamento della routine:

```
PRINT PEEK 17000 deve restituire: 62
```

```
PRINT PEEK 17001 deve restituire: 21
```

... e così via.

In pratica si può realizzare facilmente una breve routine che accetti le istruzioni in codice oggetto, le converta nell'equivalente decimale e le sistemi automaticamente nelle locazioni di memoria a partire dal byte desiderato, come il programma del listato 1.

L'utilizzazione del codice oggetto, nel quale i numeri sono sempre rappresentati da due caratteri, facilita molto il caricamento automatico. Naturalmente i numeri decimali forniti al sistema vengono ulteriormente tradotti nel loro equivalente binario prima di essere sistemati negli indirizzi di memoria.

Viene spontaneo pensare ad un programma che accetti le istruzioni direttamente in linguaggio As-

sembly e le renda comprensibili allo ZX81.

Si pensi quale vantaggio sarebbe per noi poter scrivere:

```
LD A, 15
```

che ha un preciso significato mnemonico, anzichè:

```
3A 15
```

che non ne ha alcuno. Il vantaggio appare ancora più evidente se si pensa che esistono oltre 600 diverse istruzioni nell'Assembly dello Z80.

L'Assembler è un programma che accetta istruzioni in Assembly e le rende accettabili al computer, mentre invece le routine come quelle del listato 1 possono essere definite più semplicemente CARICATORI ed accettano solo istruzioni in codice oggetto.

Con la routine LM utilizzata fino ad ora abbiamo appreso l'esistenza di alcune istruzioni, tipo LD, ADD e RET. Di tutte queste RET è certamente la più importante ed equivale al RETURN del BASIC.

In effetti ogni routine LM può essere considerata come una subroutine, che si chiama con USR, invece che con GOSUB.

Se in una subroutine BASIC si dimentica il RETURN si ottiene un messaggio di errore; nel caso del LM ciò provoca conseguenze ben più gravi, il sistema va in CRASH, e l'unica possibilità è togliere l'alimentazione.

### I registri

La figura 1 illustra in modo sommario i registri della CPU dello Z80, il microprocessore che costituisce il cuore dello ZX81.

Si considerino A, B e C; A può essere considerato una abbreviazione di *accumulatore* ed è il registro più utilizzato della CPU; infatti viene usato per l'esecuzione della mag-



## Linguaggio macchina per ZX81

L'istruzione-tipo ADD A, A (tabella 4) somma ad A il valore di A, cioè raddoppia A. Le operazioni di addizione comportano delle implicazioni per quanto riguarda il RI-PORTO.

ADD A, n somma al contenuto del registro A il valore di una costante numerica. Per esempio, se A vale 5, con l'istruzione

ADD A, 10 (esadecimale) diventa uguale a 21. Quindi è chiara la funzione di *accumulatore* svolta da A, perchè è l'unico registro che possa essere utilizzato per sommare costanti, assieme alla coppia HL.

L'istruzione-tipo ADD HL, BC somma al contenuto del registro HL il contenuto del registro BC.

Sempre nella tabella 4 si trovano codici dell'istruzione SUB, che indica la sottrazione. Quindi SUB A, A fornisce sempre zero mentre SUB A, n dà come risultato il valore di A meno il valore della costante n.

Sono disponibili meno istruzioni SUB che istruzioni ADD.

### Immagazzinamento della routine LM

Si consideri la routine LM della tabella 2 e si immagini di volerla utilizzare all'interno di un programma BASIC. Si deve procedere in questo modo:

- caricare il programma LOADER/1 (listato 1);
- dare il RUN; il programma viene eseguito e la routine LM viene sistemata nei byte dal 17000 in poi;
- cancellare il programma LOADER/1, che non serve più in quanto la routine è stata caricata, senza usare il NEW ma una linea per volta:  
10 NEWLINE  
20 NEWLINE  
... e così via
- caricare il programma BASIC che deve contenere l'istruzione UR 17000.

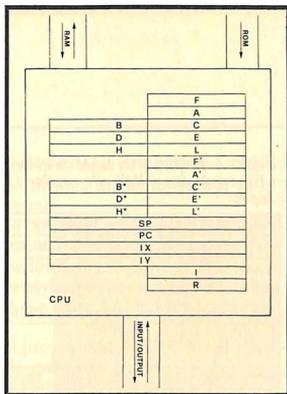


Figura 1. Illustrazione sommaria dei registri della CPU. La serie contrassegnata con l'apice è il set di registri alternativo che può essere scambiato con il set normale. Esistono due istruzioni per questo scopo, sono: EX AF, A' F' che scambia A e A' e F e F'; e EXX che scambia tutti gli altri registri.

Questo metodo è più veloce rispetto alla battitura di singole POKE, ma ha degli inconvenienti. Per esempio, se il programma BASIC è appena un poco più lungo, va ad invadere l'area degli indirizzi fino al 17000 ed oltre. In questo caso la routine LM non viene cancellata ma "slitta" letteralmente verso gli indirizzi più alti, quindi l'istruzione UR 17000 invece di incontrare la routine LM trova le ultime istruzioni del programma BASIC, e probabilmente il sistema va in CRASH.

Vi sono due alternative: chi dispone dell'espansione di memoria potrebbe immagazzinare la routine, dall'indirizzo 30000 in avanti, avendo una buona sicurezza che il programma BASIC non arriverà mai a quelle altezze, a meno che non sia particolarmente lungo.

Altrimenti se non si dispone dell'espansione ci si può premunire contro eventuali collisioni tra BASIC e LM abbassando il valore del byte RAMTOP.

Gli indirizzi 16388 e 16389 contengono l'indirizzo di RAMTOP, il primo byte non utilizzabile subito sopra la RAM, con 1 kbyte di memoria, 17408. Con le istruzioni:

POKE 16388, 104  
POKE 16389, 66

si abbassa a 17000 il valore di questo byte. Il programma BASIC non può superare questo tetto, ma rimane ancora disponibile un supporto fisico equivalente a 407 byte di memoria, dal 17001 al 17407 e qui si può collocare la nostra routine LM.

Si noti che la routine LM non subisce l'effetto del SAVE quando è situata al di sopra del BASIC, oppure sopra RAMTOP.

Se si vuole registrare e conservare la routine, questa deve essere descritta non in alto nella RAM, ma in basso assieme al programma.

Si può cambiare l'indirizzo di inizio nel programma LOADER/1, sostituendo 17000 con 16509 così la routine all'inizio della RAM; bisogna evitare che anche il programma BASIC venga scritto all'inizio della RAM sovrapprendendosi alla routine e cancellandola.

Si può caricare prima il programma BASIC ed inserire poi la routine, ma evidentemente questa collocandosi nei byte dal 16509 in poi, andrà a sovrapporsi alla parte iniziale del programma BASIC, cancellandola.

Per evitare questo inconveniente si deve scrivere una parte di programma BASIC superfluo, seguita dal programma utile; quindi si sovrappone alla parte inutile la routine LM.

Un esempio di linea inutile è:

1 REM 0

Si usa una linea REM composta almeno da tanti caratteri quanti sono i byte della routine LM.

In questo modo la routine viene immagazzinata a partire dal byte 16514 perchè i primi 5 byte sono occupati da:

2 byte per il numero di linea  
2 byte per la lunghezza della linea  
1 byte per il REM.

Si userà il programma LDC del precedente LOADER/1 (listato 2).

LDC inizia con una linea 1 REM seguita da 32 caratteri 0, e immagaz-

**Linguaggio macchina per ZX81**

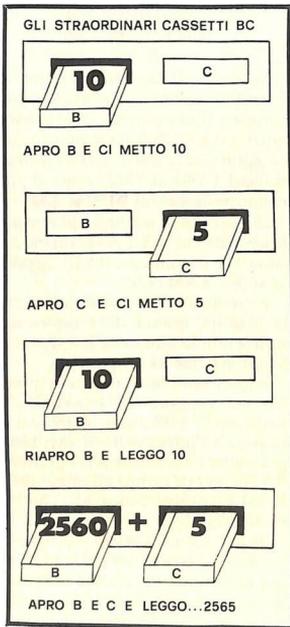


Figura 2. Anche i registri DE e HL si comportano in modo analogo a BC. B oppure C possono essere usati singolarmente per contenere numeri fino a 255, oppure in coppia per contenere numeri fino a 65535. In questo secondo caso il numero contenuto in B vale: numero \* 256. Deve essere cura del programmatore leggere ogni numero in modo corretto, ciò usando le istruzioni opportune.

zina routine LM lunghe fino a 26 byte. Si utilizzano i byte dal 16514 al 16519 per immagazzinare eventuali dati e la routine vera e propria viene memorizzata a partire dall'indirizzo 16520 e va in esecuzione con l'istruzione

USR 16520

La mappa della RAM relativa è riportata nella figura 3.

Una volta caricata la routine LM, LDC può essere cancellata esclusa

Figura 3. Mappa della RAM quando viene caricato il programma LDC. La linea 1 di LDC rende disponibili alla routine LM relativi dati i primi 37 byte, dal 16509 al 16546.

PEEK 16388 + 256 * PEEK 16389	BYTE RAMTOP	
FINE DELLA RAM	ULTIMO BYTE	AREA LIBERA
PEEK 16404 + 256 * PEEK 16405	PRIMO BYTE	AREA DELLE VARIABILI DEL PROGRAMMA
PEEK 16400 + 256 * PEEK 16401	PRIMO BYTE	AREA DELLA MEMORIA DI SCHERMO
PEEK 16396 + 256 * PEEK 16397	PRIMO BYTE	AREA DEL PROGRAMMA
	↑	INIZIO AREA DEL PROGRAMMA
	16547	
	16546	NEWLINE DELLA LINEA 1
	16545	
	↑	26 BYTE RISERVATI ALLA ROUTINE LM
	16520	
	16519	SEI BYTE CHE UTILIZZA PER IMMAGAZZINARE EVENTUALI DATI RELATIVI ALLA ROUTINE LM
	16514	
	16513	REM
	16512	LUNGHEZZA IN CARATTERI DELLA LINEA 1
	16511	
	16510	NUMERO DI LINEA DELLA LINEA 1
	16509	
	↑	125 BYTE RISERVATI ALLE VARIABILI DEL SISTEMA
	16508	
	16384	
INIZIO DELLA RAM		

```

1 REM LOADER/1
10 LET IND = 17000
20 LET H$ = "3E15C6154F0600C9"
30 FOR K = 1 TO LEN H$-1 STEP 2
40 POKE IND, 16 * CODE H$(K) + CODE H$(K + 1)-476
50 LET IND = IND + 1
60 NEXT K
70 STOP

```

Listato 1. Questo programma carica in memoria la routine LM contenuta nella stringa H\$, partendo dall'indirizzo 17000.

## Linguaggio macchina per ZX81

la linea 1, che contiene i byte dal 16514 al 16546.

Successivamente si può caricare il programma BASIC, che può essere registrato e si avvia con il RUN. Se si ha a disposizione molta memoria si può evitare di cancellare LDC, scrivendo il programma dalla linea 250 in avanti. Ovviamente dopo il primo RUN, il programma deve essere sempre avviato con GOTO 250.

### Una prova pratica

Si vuole scrivere una routine LM che legga un numero (150 decimale) collocato nell'indirizzo 16514, un altro numero (100 decimale) collocato nell'indirizzo 16515 e che li sommi,

scriva la somma nell'indirizzo 16516 e la fornisca anche in output sullo schermo.

Si utilizza il programma LDC per scrivere i due numeri (150 e 100) nelle locazioni 16514 e 16515 (riservate ai dati) e per caricare la routine LM che leggerà i numeri e li sommerà.

Il programma BASIC è costituito dalle due linee:

210 PRINT USR 16520

220 PRINT PEEK 16516

La linea 210 manda in esecuzione la routine LM e restituisce sullo schermo il contenuto di BC, cioè la somma di 150 e 100. La routine scrive il risultato anche nella locazione 16516 e con la linea 220 permette di verificare che ciò sia veramente ac-

caduto.

La tabella 5 illustra lo svolgimento di questo programma.

Si deve procedere nel seguente modo:

si carica (conviene conservarlo sempre pronto all'inizio di un nastro) e si aggiungono le linee 210 e 220 senza cancellare LDC.

Dopo aver dato il RUN si inseriscono tutti i numeri esadecimali della colonna "Codice Oggetto", si possono scrivere prima tutti e battere poi NEWLINE, oppure uno alla volta, o anche in gruppi:

9664	NEWLINE
0000000	NEWLINE
3A8240	NEWLINE
...	e così via.

# Per 'lavorare' al meglio con il Pet e l'M20

## Paolo e Carlo Pascolo

# IL BASIC DEL PET E DELL'M20

Il personal computer rappresenta oggi, oltre che un valido aiuto nel lavoro, anche un'irresistibile tentazione. Può capitare, così, che qualcuno si trovi a disporre di un Commodore o di un M 20 Olivetti senza conoscerne appieno il linguaggio e le possibilità. Questo volume vuol rappresentare proprio un prezioso supporto per chi debba, o voglia imparare a programmare in Basic su questi strumenti di lavoro, gioco o studio: comandi, istruzioni, informazioni, consigli... fino a diventare davvero "padroni" di due dei più diffusi Personal Computer.

**226 pagine. Lire 16.000**  
**Codice 336 D**

Per ordinare il volume  
utilizzare l'apposito tagliando  
inserito in fondo alla rivista

**SCONTO 20% AGLI ABBONATI  
FINO AL 28-2-84**



**GRUPPO EDITORIALE  
JACKSON**



## Linguaggio macchina per ZX81

Terminando l'operazione con il codice C9, si preme S per indicare il terminale della routine. Il computer visualizza un output come quello della figura 4, dal quale si può verificare se la routine è stata inserita correttamente.

La prima colonna rappresenta gli indirizzi, la seconda il codice esadecimale inserito, la terza l'equivalente decimale e la quarta il carattere equivalente al numero della terza colonna; gli equivalenti decimali non sono altro che i numeri di codice (CODE) del carattere immagazzinato in quell'indirizzo. Evidentemente il computer interpreta il contenuto dell'indirizzo come una istruzione LM, ma per comodità, usando la funzione CHR\$ si può ricavare il carattere alfanumerico o grafico equivalente.

Dando il list si nota che nella linea 1 molti zeri sono stati trasformati nel carattere equivalente al nuovo valore immagazzinato in quel byte.

La linea 1 in effetti potrebbe costituire un ottimo LIST della routine LM, se non vi fossero alcuni errori introdotti dal sistema. Per esempio quando si usa il codice 7E, spariscono 6 caratteri. Inconvenienti si possono verificare anche con il codice 76, ed inoltre tutti i codici dal 67 al 126 decimale vengono invariabilmente rappresentati con un punto interrogativo.

La linea 1 presenta però il vantaggio di poter essere posta in basso nello schermo in modo EDIT; si può quindi modificare la routine LM aggiungendo o togliendo alla linea 1 i rispettivi caratteri CHR\$. Non si toglia mai il TAN finale, che equivale al codice decimale 201, cioè all'istruzione RET.

Ritornando alla figura 4, se al controllo riscontrate qualche errore potete correggerlo con una POKE: per esempio se leggete:

16521      83      131  
anziché

16521      82      130

si può correggere l'errore scrivendo:  
POKE 16521, 130

Figura 4. Una volta terminato il caricamento della routine LM, il programma LDC offre una videata di controllo come quella illustrata in figura, comprendente: indirizzo della locazione, codice oggetto, equivalente decimale ed equivalente CHR\$. Eventuali correzioni possono essere fatte con istruzioni POKE rivolte agli indirizzi sbagliati. La routine visualizzata è quella della tabella 5.

16514	96	150	□
16515	64	100	?
16516	00	0	
16517	00	0	
16518	00	0	
16519	00	0	
16520	3A	58	U
16521	82	130	L
16522	40	64	RND
16523	57	87	?
16524	3A	58	U
16525	83	131	■
16526	40	64	RND
16527	82	130	L
16528	32	50	M
16529	83	131	■
16530	40	64	RND
16531	4F	79	?
16532	06	6	■
16533	00	0	
16534	C9	201	TAN

```

1 REM 00000000000000000000000000000000
0000000000
2 REM LDC
10 LET H$=""
10 LET H$="16514
20 IF H$="" THEN PRINT AT 0,0;
IND; " (IS=STOP)";
30 IF H$="S" THEN INPUT H$
40 POKE IND,ASC CODE H$+CODE H$
(2)-476
50 LET IND=IND+1
60 GOTO 30
100 CLR
110 FOR K=16514 TO IND-1
120 LET H$=""
130 FOR US=1 TO 2
140 LET H$=CHR$(2B+N-16*INT(N
/16)+H$
150 LET N=INT(N/16)
160 NEXT U
170 PRINT K;TAB 7;H$;TAB 11;PEE
K
180 NEXT K
190 FOR K
200 PRINT USR 16520
210 PRINT

```

Listato 2. Questo programma può essere usato per caricare routine LM in codice esadecimale. È un po' lungo ma molto efficace. Le linee dalla 100 alla 190 consentono di listare la routine introdotta. LDC è un'abbreviazione di LOAD AND CHECK (carica e controlla). La figura 4 illustra la videata di controllo offerta da LDC.

```

1 REM 00000000 RND? TAN
2 REM MINIPAC
100 LET H$=0
110 LET T=0
120 FOR K=1 TO 6 STEP .02
130 LET D=R
140 PRINT AT 0,H;
150 IF USR 16520=100 THEN LET T
+1
160 IF USR 16520=21 THEN GOTO 3
00
200 PRINT CHR$(128)
205 PAUSE 30
210 LET H$=H+(INKEY$="O" AND H)+
(INKEY$="D" AND H/16)
220 PRINT AT 14,RND*15;CHR$(21
230 LET D=D+15;INT K
240 IF D>15 THEN GOTO 220
250 PRINT AT 14,RND*15;CHR$(199
260 SCROLL
270 SCROLL
280 NEXT K
290 PRINT "PUNT. ";T
310 GOTO 1

```

Listato 3. In Minipac il giocatore muove prendendo 0 oppure P. Deve evitare le + e deve mangiare le X.

oppure, se nessun carattere 7E o 76 ha apportato cambiamenti, è possibile editare la linea 1 cancellando il carattere CHR\$(131) e inserire al suo posto il carattere CHR\$(130). Quindi si verifica l'avvenuta correzione con il comando: POKE 16521 o meglio GOTO 100.

## Linguaggio macchina per ZX81

Chi dispone del sistema da 1 Kbyte non ottiene la routine visualizzata per intero; si usi CONT per la parte mancante.

Terminata la verifica, per mandare in esecuzione le linee BASIC si preme ancora CONT, ottenendo:

```
250
250
```

In conclusione una nota importantissima sull'ordine di memorizzazione dei numeri.

L'istruzione LD A(4082) carica nel registro A il contenuto dell'indirizzo 16514, cioè 150 decimale.

4082 è l'equivalente esadecimale di 16514 e deve essere scritto nelle locazioni 16521 e 16522 (vedi la tabella 5). Si noti che il byte 16521 deve contenere la parte *meno significativa* di 4082, (cioè 82) mentre il byte 16522 quella *più significativa*. Quindi il codice oggetto di LD A, (4082) diventa:

```
3A 82 40
```

dove i due byte 40 e 82 risultano invertiti.

L'uso della parentesi ha un preciso significato; l'istruzione:

```
LD A,4082
```

non avrebbe senso perché non si potrebbe caricare nell'accumulatore il numero 16514, dato che è maggiore di 255. L'istruzione:

```
LD A,(4082)
```

indica invece che vogliamo caricare nell'accumulatore il contenuto del byte 16514; i numeri tra parentesi sono sempre riferiti ad un indirizzo di memoria.

### Alcuni programmi

I listati 3, 4, 5 e 6 utilizzano tutti la routine LM illustrata nella tabella 6. Questa routine legge il numero contenuto negli indirizzi 16398 e 16399, che corrisponde ad un indirizzo contenente il codice del carattere occupante la posizione di stampa.

Quando il giocatore muove, sposta la sua "pedina" in una certa posizione. Prima che lo spostamento avvenga, il computer legge con la routine LM il carattere che sta per

I listati 3, 4, 5 e 6 girano solo se prima viene caricata in memoria una routine LM utilizzando il programma LDC come descritto nell'articolo. Chi non è interessato al linguaggio macchina può risolvere il problema con:

```
1 REM 00000E:RND? TAN
```

I tredici caratteri sono: zero/zero/zero/zero/zero/lettera E/due punti/shift, function e RND/punto interrogativo/shift, graphics e T/spazio/shift, function e TAN.

Si deve anche eseguire POKE 16523,78. In questo modo la routine LM è sistemata in memoria e si può caricare uno dei quattro listati a scelta, dalla linea 2 in poi. Le linee PAUSE (205 per Minipac, 136 per Carrera e Gulp, 160 per Sbruzz) servono solo sullo ZX80/8 Kbyte e devono essere cancellate da chi usa lo ZX81. Se il gioco diventa troppo veloce si può rallentare inserendo al posto della linea PAUSE cancellata, un ciclo di ritardo; per esempio:

```
FOR U = 1 TO 300
```

```
NEXT U
```

Tabella 1. Byte usati dallo ZX81 per immagazzinare programmi BASIC.

BYTE OCCUPATI NELL'AREA DEL PROGRAMMA	
PER OGNI LINEA DI PROGRAMMA, IN OGNI CASO:	5 BYTE FISSI: 2 PER IL NUMERO DI LINEA, 2 PER IL CONTROLLO DELLA LUNGHEZZA DELLA LINEA 1 PER IL NEWLINE.
PER QUALSIASI CARATTERE, ESCLUSI I NUMERI POSTI FUORI DALLE STRINGHE	1 BYTE
PER OGNI NUMERO POSTO FUORI DALLE STRINGHE, ED ESCLUSI I NUMERI DI LINEA	6 BYTE PIU' UN BYTE PER OGNI CIFRA DEL NUMERO.
PER OGNI ISTRUZIONE, CARATTERE GRAFICO, FUNZIONE, ECC.	1 BYTE
<i>NELLA MEMORIA DI SCHERMO: 793 BYTE FISSI CON L'ESPANSIONE INSERITA, DA 25 A 793 SENZA L'ESPANSIONE.</i>	
BYTE OCCUPATI NELL'AREA DELLE VARIABILI DEL PROGRAMMA	
PER OGNI VARIABILE STRINGA	3 BYTE PIU' UN BYTE PER OGNI CARATTERE TRA GLI APLICI.
PER OGNI VARIABILE NUMERICA	5 BYTE PIU' UN BYTE PER OGNI CARATTERE DEL NOME DELLA VARIABILE.
PER OGNI MATRICE NUMERICA	4 BYTE PIU' 2 BYTE PER OGNI DIMENSIONE PIU' 5 BYTE PER OGNI ELEMENTO.
PER OGNI VARIABILE DI CONTROLLO DEI CICLI FOR NEXT	18 BYTE

Tabella 2. Esempio di routine LM. Questa routine viene caricata in memoria con il programma LOADER/1.

INDIRIZZI	CODICE ASSEMBLY O OPCODES	CODICI OGGETTO O EXCODES	EQUIVALENTE DECIMALE	ANALOGIA BASIC
17000	LD A, 15	3E	62	LET A = 21
17001		15	21	
17002	ADD A, 15	C6	198	LET A = A + 21
17003		15	21	
17004	LD C, A	4F	79	LET C = A
17005	LD B, 0	06	6	LET B = 0
17006		00	0	
17007	RET	C9	201	RETURN

## Linguaggio macchina per ZX81

### Immagazzinamento di numeri in due byte

Ogni byte è composto da 8 bit, cioè otto informazioni elementari, che possono avere due soli valori: 1 oppure 0. Qualsiasi quantità di 1 o 0 combinati fra loro costituisce un numero binario. Per esempio:

110011

equivale al numero 51.

Anche i byte immagazzinati nella memoria dello ZX81 vengono letti come numeri binari; usando un byte potremo rappresentare tanti numeri diversi, quante sono le combinazioni possibili degli otto bit che lo costituiscono. Tra la combinazione più bassa: 00000000 e la combinazione più alta: 11111111, esistono 254 combinazioni intermedie, quindi un byte può rappresentare 256 numeri diversi: tutti quelli che vanno da 0 a 255. I due precedenti numeri possono essere raffigurati in questo modo, tenendo conto che i bit si contano partendo dal primo a destra:

8° bit	0	0*	128	=	0+	1	1	1*	128	=	128 +
7° bit	0	0*	64	=	0+	1	1	1*	64	=	64 +
6° bit	0	0*	32	=	0+	1	1	1*	32	=	32 +
5° bit	0	0*	16	=	0+	1	1	1*	16	=	16 +
4° bit	0	0*	8	=	0+	1	1	1*	8	=	8 +
3° bit	0	0*	4	=	0+	1	1	1*	4	=	4 +
2° bit	0	0*	2	=	0+	1	1	1*	2	=	2 +
1° bit	0	0*	1	=	0	1	1	1*	1	=	1 =
			0						256		

Volendo immagazzinare il numero 15 nel byte 16800, dobbiamo scrivere:

POKE 16800, 15

e lo ZX81 sistema nel byte 16800 la traduzione binaria di 15, cioè:

00001111

Ammettiamo ora di voler immagazzinare il numero 400. Una soluzione possibile sarebbe di sistemare 255 nel byte 16800 e 145 nel byte 16801, per cui la somma dei contenuti dei due byte fornirebbe 400. Ma in questo modo, volendo immagazzinare 2500 avremmo bisogno di 10 byte.

Ci comportiamo allora in modo diverso. Mettiamo il numero 144 nel byte 16800, ed il numero 1 nel byte 16801. Però l'uno contenuto in 16801 non vale 1, bensì: una volta 256.

Volendo immagazzinare il numero 16509, dobbiamo scrivere:

POKE 16800, 125

POKE 16801, 64

e per leggerli dobbiamo scrivere:

PRINT PEEK 16800

PRINT PEEK 16801 \* 256

cioè: PRINT PEEK 16800 + 256 \* PEEK 16801.

che equivale a: SCRIVI 125 + 256 \* 64.

Con questo metodo possono essere immagazzinati numeri fino A:

255 + 256 \* 255 = 65535

Tabella 3. Codici esadecimali dell'istruzione LD REG, REG.

LD	A	B	C	D	E	H	L
A	7F	78	79	7A	7B	7C	7D
B	47	40	41	42	43	44	45
C	4F	48	49	4A	4B	4C	4D
D	57	50	51	52	53	54	55
E	5F	58	59	5A	5B	5C	5D
H	67	60	61	62	63	64	65
L	6F	68	69	6A	6B	6C	6D

```

1 REM 000000E:RND?# TAN
2 REM CRARRA 1K
100 LET C=10
105 LET P=0
110 PRINT "C: C";
120 IF USR 16520=102 OR USR 165
130 PRINT "0":GOTO 300
135 P=P+1
140 SCROLL P*1
145 LET H=INT (RND*2)
150 IF RND>.49 THEN LET H=-H
155 LET V=Y+H
160 IF V<0 THEN LET V=0
165 IF V<8 THEN LET V=8
170 LET C=C+(INKEY$="D" AND 1) +
(INKEY$="P" AND 1)
180 PRINT AT 1,C,"V:V";TAB 5+4*K
300 PRINT "CRASH"
305 PRINT "TAB C:P"
310 STOP
    
```

Listato 4. In Carrera il giocatore guida la sua auto lungo una strada senza fine. Il gioco termina solo quando si verifica il CRASH. Per muovere premere 0 e P.

```

1 REM 000000E:RND?# TAN
2 REM GULD 1K
100 LET Y=0
105 LET X=TO -.5 STEP -.02
110 PRINT AT 0,C,"X";
120 IF USR 16520=120 THEN GOTO 300
130 PRINT "X"
135 SCROLL "X"
140 PRASE 30
145 LET H=INT (RND*2)
150 IF RND>.49 THEN LET H=-H
155 LET V=Y+H
160 IF V<0 THEN LET V=0
165 IF V<8 THEN LET V=8
170 LET C=C+(INKEY$="D" AND 1) +
(INKEY$="P" AND 1)
180 NEXT K
185 PRINT AT 1,C,"X"
190 PRINT AT 2,Y,"Y";TAB 5+4*K
200 STOP
210 PRINT "CRASH.. "
300 IF INKEY$="0" THEN PRASE 200
305 PRASE 100
310 CLS
320 RUN
    
```

Listato 5. In questo gioco la strada si restringe sempre più. Riuscirà il giocatore a guidare la sua auto fino al termine evitando il CRASH..? Per muovere premere 0 e P.

Listato 6.

```

1 REM 000000E:RND?# TAN
100 LET X=PI*PI
110 LET C=X
120 LET P=PI*PI
130 PRINT AT 1,C,"S"
140 PRASE 10
145 LET H=INT (RND*2)
150 IF USR VAL "16520"=VAL "0"
160 PRINT CHR$(VAL "120")
170 PRINT AT 1,C,"S"
180 PRINT AT 2,C,"S"
190 IF RND>VAL ".75" THEN LET H
="H"
200 LET L=L-H
210 LET L=VAL "0" OR L>VAL "7" T
220 IF C<VAL "0" OR C>VAL "14"
230 THEN PRINT AT 1,C,"S"
240 PRINT AT 1,C,"S"
250 PRINT AT 1,C,"S"
260 IF USR VAL "16520"=VAL "120"
270 THEN GOTO VAL "180"
280 PRINT AT 2,C,"S"
290 PRINT AT X,Y,"H"
300 PRINT AT X,Y,"H" AND 1 +
(INKEY$="0" AND 1)
310 PRINT AT X,Y,"H" AND 1 +
(INKEY$="S" AND 1)
320 GOTO VAL "130"
330 PRINT AT X,Y,"S";
340 IF P=INT (RND*VAL "50")
350 PRINT AT VAL "20",VAL "S";P
    
```



## Linguaggio macchina per ZX81

Tabella 4. Codici esadecimali di alcune istruzioni Assembly Z80.

ASSEMBLY	CODICE OGGETTO	ASSEMBLY	CODICE OGGETTO	ASSEMBLY	CODICE OGGETTO
LD A, n	3E	ADD A, A	87	SUB A, A	97
LD B, n	06	ADD A, B	80	SUB A, B	90
LD C, n	0E	ADD A, C	81	SUB A, C	91
LD D, n	16	ADD A, D	82	SUB A, D	92
LD E, n	1E	ADD A, E	83	SUB A, E	93
LD H, n	26	ADD A, H	84	SUB A, H	94
LD L, n	2E	ADD A, L	85	SUB A, L	95
LD BC, nn	01	ADD HL, BC	09		
LD DE, nn	11	ADD HL, DE	19		
LD HL, nn	21	ADD HL, HL	29		
RET	C9	ADD A, n	C6	SUB A, n	D6

```

385 LET L=VAL "10"
387 LET C=PI-P1
388 PAUSE 80
389 IF INKEY<:"" THEN PAUSE 30
390 PRINT AT 1:1:1
391 GOTO URL "16E2"
  
```

Listato 6. Un vermicciatolo zampetta sullo schermo ed il giocatore deve schiacciarlo. Ogni volta che ci riesce fa uno Sbruzz. Le particolari tecniche usate nella stesura di questo programma consentono un notevole risparmio di memoria, infatti gira anche su ZX81 senza espansione; si raccomanda un'analisi riferendosi alla tabella 1. PI intende Pigreco. (SHIFT, FUNCTION e M). Muovere premendo 5, 6, 7 e 8.

# Spectrum

con  
"supergaranzia originale"



Se volete riceverlo velocemente compilate e spedite in busta il "Coupon Sinclair" e riceverete in OMAGGIO il famoso libro "Guida al Sinclair ZX Spectrum" di ben 320 pagine, del valore di L. 22.000.

## EXELCO

Via G. Verdi, 23/25  
20095 - CUSANO MILANINO (MILANO)

Descrizione	Qt.	Prezzo unitario	Totale L.
Personal Computer ZX Spectrum 16K RAM con alimentatore, completo di manuale originale Inglese e cavetti di collegamento.		L. 299.000	
Personal Computer ZX Spectrum 48K RAM con alimentatore, completo di manuale originale Inglese e cavetti di collegamento.		L. 399.000	
Kit di espansione 32K RAM		L. 99.000	
Stampante ZX Printer		L. 180.000	
Guida al Sinclair ZX Spectrum		L. 22.000	
Cassetta programmi dimostrativi per il rapido apprendimento alla programmazione e utilizzo dello ZX Spectrum in Italiano.		L. 48.000	

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

Nome

Cognome

Via

Città

Data    C.A.P.

Desidero ricevere la fattura  SI  NO

Partita I.V.A. o, per i privati Codice Fiscale

Acconto L.

Sarà data precedenza alle spedizioni, se assieme all'ordine verrà incluso un anticipo di almeno L. 10.000. Aggiungere L. 5.000 per il recapito a domicilio. I prezzi vanno maggiorati dell'I.V.A. 18% e sono validi fino a giugno 1984.



Tabella 5. *Illustrazione di una routine LM che somma due numeri contenuti nei byte 16514 e 16515 ponendo il risultato nel byte 16516. La routine viene caricata con il programma LDC.*

INDIRIZZO	ASSEMBLY	CODICE OGGETTO	EQUIVAL. DECIMALE	ANALOGIA BASIC	COMMENTO
16514		96	150	POKE 16514, 150	<i>SCRIVE NEI PRIMI INDIRIZZI I DUE NUMERI DA SOMMARE</i>
		64	100	POKE 16515, 100	
		00	0	POKE 16516, 0	
		00	0	POKE 16517, 0	
		00	0	POKE 16518, 0	
		00	0	POKE 16519, 0	
16520	LD A, (4082)	3A	58	LET A = PEEK 16514	<i>QUI INIZIA LA ROUTINE LM. CARICA IN A IL NUMERO CONTENUTO NEL BYTE 16514</i>
		82	130		
		40	64		
16523	LD D, A	57	87	LET D = A	<i>SISTEMA A NELLA VARIABILE D</i>
16524	LD A, (4083)	3A	58	LET A = PEEK 16515	<i>CARICA IN A IL NUMRO CONTENUTO NEL BYTE 16515</i>
		83	131		
		40	64		
16527	ADD A, D	82	130	LET A = A + D	<i>ESEGUE LA SOMMA</i>
16528	LD (4083), A	32	50	POKE 16516, A	<i>SCRIVE LA SOMMA NEL BYTE 16516</i>
		83	131		
		40	64		
16531	LD C, A	4F	79	LET C = A	<i>CARICA I REGISTRI BC</i>
16532	LD B, 0	06	6	LET B = 0	
		00	0		
16534	RET	C9	201	RETURN	<i>FINE DELLA ROUTINE</i>
FINO AL 16545					<i>NON USATI</i>

Tabella 6. *Illustrazione di una routine che fornisce il codice del carattere occupante la posizione di stampa. La routine viene caricata con il programma LDC.*

INDIRIZZO	ASSEMBLY	CODICE OGGETTO	EQUIVAL. DECIMALE	ANALOGIA BASIC	COMMENTO
16514		00		POKE IND, 0	<i>BYTE RISERVATI AI DATI, NON USATI</i>
		00			
		00			
		00			
		00			
		00			
16520	LD HL, (400E)	2A	42	LET HL = PEEK 16398 + 256 ★ PEEK 16399	<i>LEGGE L'INDIRIZZO CONTENUTO NEI BYTE 16398 E 16399 E LO METTE IN HL</i>
		0E	14		
		40	64		
16523	LD C, (HL)	4E	78	LET C = PEEK HL	<i>LEGGE IL CONTENUTO DEL BYTE HL</i>
16524	LD B, 0	06	6	LET B = 0	
		00	0		
16526	RET	C9	201	RETURN	
16527 16545					<i>NON USATI</i>



## Linguaggio macchina per ZX81

Tabella 7. Può essere usata per la conversione rapida decimale/esadecimale di numeri fino a 255 (decimale). Linee e colonne nel loro punto di intersezione forniscono l'equivalenza.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

essere soprastampato dalla pedina del giocatore. Nel caso del listato 3, se il carattere è un segno + la partita finisce, se invece è un segno X il punteggio del giocatore viene incrementato.

Per tutti e quattro i listati si deve caricare LDC, ed inserire la routine LM della tabella 6:

```
000000000000
2A0E40
4E
0600
C9
```

Per evitare gli zeri iniziali si può cambiare la linea 10 di LDC con: LET IND = 16520.

Successivamente si cancelli LDC una linea per volta ed si inserisca a scelta il listato 3, 4, 5 o 6; questi programmi possono essere registrati e possono essere avviati con il RUN.

# non perdetevi il nuovo numero di

- Bitest: Alphatronic PC
- Personal Computer Show '83
- SICOB '83
- TEXAS CC40
- Atari Scram:  
un gioco di simulazione
- Screenwriter II
- Riservato Personal



## Quando un "piccolo" funziona come un "grande"

di D. Di Mario

**L**o ZX81 ed il più recente Spectrum hanno una precisione di 9 cifre e mezzo e un campo di numeri che va da circa  $10^{-39}$  a  $10^{38}$ .

Queste caratteristiche vanno bene per una vasta gamma di applicazioni ma presentano dei limiti nel campo prettamente scientifico.

Un caso pratico si è verificato quando si è trattato di controllare numericamente una teoria di unificazione tra le forze elettriche e quelle gravitazionali. Il presupposto era di considerare l'elettrone come un buco nero oscillante e analizzarne il comportamento.

Lo studio fu iniziato molti anni fa presso una università straniera ma solo recentemente si è riusciti a concretizzare il tutto in un programma di calcolo.

Il problema era di vedere se con 3 solo dati iniziali, cioè: G, la costante di gravitazione universale che vale  $6,67302 \cdot 10^{-11}$ ; c, la velocità della luce che vale  $299792458$  e h la costante di Planck che vale  $6,62617 \cdot 10^{-34}$ , si potevano ricavare tutte le costanti elettriche inclusa la massa e carica dell'elettrone.

Pur non essendo particolarmente complessi, i calcoli richiedono, almeno in un caso, una precisione di 10 cifre e mezzo ed un elevamento a potenza che poteva arrivare fino a 84. Quindi si doveva appurare se era

possibile trasformare il personal in un vero e proprio "number cruncher".

Occorre dire che i risultati sono stati totalmente positivi anche se il programma impiega ben 5 minuti, in modo FAST, per ottenere i risultati. Se si tiene conto che i dati iniziali possono variare un poco (pur rimanendo nei limiti della tolleranza) si può immaginare quanto tempo ci vuole per ottenere delle correlazioni.

Il problema più frequente, che avrebbe normalmente provocato il blocco del personal con codice di errore B, è l'elevamento a potenza che durante un calcolo può superare, anche se temporaneamente, il campo di funzionamento dello ZX.

Questo è anche il problema più facile da risolvere perché è sufficiente ristimare l'ordine delle operazioni in una formula, assicurandosi, possibilmente con l'aiuto di un calcolatore tascabile, che non si eccedano i limiti dello ZX.

Per esempio, la formula per il calcolo della costante di gravitazione è la seguente:

$$G = \frac{c}{\pi h} (2a - a^2)^2 \left( \frac{c}{2\pi} \frac{e}{2\pi} \right)^4$$

dove è la costante di struttura fine e vale  $0,0072973538$  ed è la carica dell'elettrone e vale  $1,6021895 \cdot 10^{-19}$ .

Il blocco si sarebbe verificato subito perché  $c/\pi h$  già eccede i limiti dello ZX81. Riorganizzando i fattori come mostrato nella linea 447 e 450 del listato 1 si è risolto il problema.

Si può notare che si evita l'elevamento a potenza dove ciò è pratico per migliorare la precisione dei risultati. Anche le radici quadrate provocano un abbassamento della precisione e quindi vanno usate solo dove strettamente necessarie.

E se il risultato finale eccede i limiti del Sinclair?

In tale caso serve a poco riarrangiare i termini della espressione, perché, alla fine, il risultato eccederà in ogni caso i limiti e produrrà il codice di errore B.

In queste circostanze occorre dividere l'espressione per un numero fisso (se il risultato finale sarà troppo grande) o moltiplicare per un numero fisso se il risultato finale sarà troppo piccolo. Si riporta così il risultato nel campo di funzionamento del personal.

Occorre però annotare da qualche parte che si è moltiplicato o diviso per un certo numero e il comando REM torna utile proprio per questo scopo.

Si è applicata questa procedura per il calcolo di HH che rappresenta la potenza di un buco nero in espansione alla velocità della luce e che vale  $7,2578906 \cdot 10^{22}$  e per il calcolo di S, il cui inverso è quantificazione ultima del tempo e vale  $4,1752626 \cdot 10^{42}$ .

Il problema più difficile ma anche il più interessante è stato quello di estendere il campo di precisione del Sinclair.

Un dato importante da calcolare era la velocità di rotazione dell'elettrone su se stesso; questo dato che come valore è prossimo alla velocità della luce, entra nel calcolo di tutte le altre costanti e deve essere valutata con precisione centimetrica. La velocità risultante è infatti di  $299245035,25$  m/sec.

Alla linea 215 viene impostato un ciclo FOR ... NEXT in cui la velocità di rotazione iniziale V viene incrementata di cm in cm finché si verificano certe condizioni che permettono di uscire dal ciclo e procedere alla stampa dei risultati.

Per estendere la precisione si è

**ZX81**  
Superscientifico

sottratto una quantità fissa da V che è la velocità che vogliamo sia valutata in alta precisione.

In questo modo togliamo delle cifre all'inizio del numero e estendiamo il numero verso destra, cioè verso una precisione maggiore del numero stesso.

Naturalmente, questa quantità che si è tolta deve essere presa in considerazione durante i calcoli seguenti.

Per esempio, A deve essere valutata con la massima precisione e la formula è la seguente:

$$A = 2 \left( \frac{c^2 - u^2}{c^2} \right)$$

Per rendere le cose semplici si è riscritta l'espressione come segue:

$$A = 2 \frac{(c - u)(c + u)}{c^2}$$

Questa formula si presta ad essere corretta senza alterare il valore di U che conserva così la sua precisione.

Nella sottrazione basta togliere a c la stessa quantità tolta ad u per ottenere lo stesso risultato. La sottrazione è l'operazione più critica perchè i due numeri sono molto vicini l'uno all'altro; la somma non è molto critica ed è sufficiente ripristinare il valore originale di u. L'espressione finale scritta nel programma è dunque la seguente:

$$A = \frac{2(c - 2,9E8 - u)(c + u + 2,9E8)}{c^2}$$

Tenendo conto degli accorgimenti suddetti si è potuta raggiungere una precisione che si può trovare solo su calcolatori di costo elevato.

Poichè il tempo di esecuzione del programma è di 5 minuti, è stato providenzialmente inserito un co-

mando di COPY che copiando lo schermo, come in figura 1, produce rumore, avvisando che l'esecuzione è terminata e si può lasciare il caffè che si stava sorseggiando.

Per chi si interessa di queste cose, viene elencato una breve descrizione delle variabili non menzionate nel testo (in ordine di apparizione nel programma).

EE = 2766602400 - Quanto fondamentale di energia.

MS = 6,2899508E13 - Massa massima.

MC = 3,0782603E-8 - Massa media.  
MM = 1,5064803E-29 - Massa minima.

O = Resistenza del vuoto (1° valore).

Q = 2,6481427E-18 - Carica elettrica statica.

QS = Carica elettrica massima equivalente.

A = Struttura fine (1° valore).  
V = 29979245031,1 - Velocità iniziale di rotazione.

QS = 2,6476004E-18 - Carica elettrica statica equivalente.

QX = 2,6524438E-18 - Carica virtuale dell'elettrone.

OM = 376,73032 - Resistenza del vuoto (valore finale).

E = 1,6021895E-19 - Carica reale dell'elettrone.

K = 9,1095398E-31 - Massa dell'elettrone.

M = 9,1095396E-31 - Massa di riferimento.

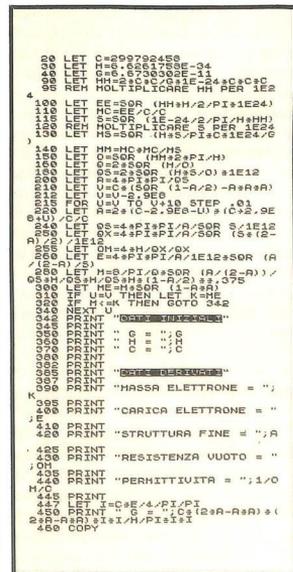
L'analisi dei dati attraverso lo ZX 81 è fondamentale per un calcolo di questa portata poichè sarebbe al limite delle possibilità umane pretendere di farle a mano.

Adesso non rimane altro che valutare le implicazioni pratiche che sono in fase di studio. Un primo successo si è avuto con il calcolo accurato della costante G il cui valore apparve già qualche anno fa (Wi-

reless World, Dicembre 1980) ed è stato confermato in pieno dalle ultimissime e più accurate misure sperimentali.



Figura 1. Risultato finale prodotto dal programma.



Listato 1. Il listato BASIC.

# I file su cassetta nel TI99/4A

## Qualche utile suggerimento per organizzare i vostri archivi

di Filippo Cerulo

**L**a gestione dei file su cassetta nel TI99/4A, rappresenta, forse, una delle maggiori difficoltà che incontrano molti utenti, soprattutto principianti.

I motivi di ciò vanno ricercati soprattutto nella impostazione particolare del manuale che accompagna la macchina: la sezione che riguarda i file è chiarissima dal punto di vista tecnico, (basta leggere con attenzione), ma è assolutamente priva di esempi di utilizzo, cosicché una persona che ignori il significato e l'uso di file e record dopo mezza pagina salta magari al capitolo sulla grafica che è senz'altro più accessibile. Inoltre, dai programmi pubblicati su alcune riviste, potrebbe dedursi che anche utilizzatori più smaliziati hanno qualche dubbio su questo importante argomento. Allora forse sarà utile spendere qualche parola in più.

### File e record

La traduzione italiana più frequente della parola file è archivio.

Infatti un file non è altro che un archivio, cioè un insieme di informazioni logicamente correlate, che in genere vengono conservate per successive consultazioni o aggiornamenti.

Ci sono infiniti esempi di archivio, anche molto vicini alla nostra esperienza comune. Sono archivi: l'agenda degli indirizzi, il libro delle ricette, l'estratto conto della banca ecc. Per fissare le idee, consideriamo un esempio banale: una rubrica telefonica.

Essa non è altro che un elenco di nomi e numeri telefonici come per esempio:

ROSSI MARCO	06/871190
VERDI ELIO	0823/771432
BIANCHI ALEX	082/919293

Se identifichiamo, come in questo caso, il file con una tabella costituita da tante righe contenenti informazioni dello stesso tipo, definiamo RECORD LOGICO del file, ogni riga della tabella.

Nel nostro caso un record logico è per es.: VERDI ELIO 0823/771432.

Ogni record è formato a sua volta da CAMPI. I record del nostro archivio sono formati da tre campi:

CAMPO N.	NOME DEL CAMPO	CONTENUTO
1	"cognome"	Verdi Elio
2	"nome"	Elio
3	"numero"	0823/771432

Le operazioni che normalmente si compiono su un archivio (la nostra rubrica telefonica) sono:

- 1) *Creazione dell'archivio*: quando copiate sull'agenda nuova i numeri che volete conservare.
- 2) *Aggiunta di un record*: quando scrivete il numero di quella tedesca bionda conosciuta al mare.
- 3) *Cancellazione di un record*: quando decidete che Carlo è veramente antipatico ed il suo numero è meglio non averlo.
- 4) *Modifica di un record*: quando il vostro medico di fiducia cambia casa e quindi numero di telefono.

In generale queste sono anche le operazioni che si compiono su qualunque tipo di archivio. L'organizzazione di un archivio dipende in parte dalle informazioni che esso deve contenere, in parte dalle operazioni che volete compiere sui record che lo costituiscono. Ad esempio, i record logici della nostra rubrica telefonica potrebbero anche avere solo due campi:

CAMPO N.	NOME DEL CAMPO	CONTENUTO
1	"nome"	Verdi Elio
2	"telefono"	0823/771432

Ciò vuol dire che una volta scelte le informazioni che dobbiamo archiviare, potremo organizzarle in record di tipo diverso a seconda delle nostre esigenze.

Un'altra operazione fondamentale che di solito si compie sui file è l'ordinamento. Esso in generale può essere fatto secondo qualunque campo del record. Ad esempio potremo ordinare i nomi della nostra rubrica alfabeticamente oppure per numeri telefonici crescenti.

Esistono svariati metodi per ordinare un file, ma su di essi non mi soffermerò. Mi preme solo ribadire che il file è una struttura usata soprattutto per conservare le informazioni a computer spento e come tale ha un'importanza fondamentale. Vi consiglio perciò di approfondire l'argomento su testi specifici o su un buon manuale di programmazione. In ogni caso altri esempi saranno esaminati in seguito.

### Il TI99/4A e i file

A differenza di molti altri computer della sua classe, il nostro gestisce i file in modo notevolmente sofisticato.

**I file su cassetta nel TI99/4A**

cato, sicuramente all'altezza di macchine più potenti e costose. Soprattutto con i DISK DRIVE offre possibilità molto ampie, tali da garantire una corretta e facile gestione di grandi masse di dati.

Nel seguito però non si parlerà dei DISK DRIVE (alzi la mano chi ne possiede uno) ma del più semplice (ma non meno utile) registratore a cassette. Tornando al nostro esempio, la rubrica telefonica, supponiamo di voler conservare i numeri di dieci amici che abbiamo memorizzato in due vettori di dieci elementi: NOM\$(I) e TEL\$(I). Il puntatore I indica la corrispondenza tra nomi e numeri. Così, ad esempio, per I = 3 si avrà:

```
NOM$(3) = BIANCHI ALEX
TEL$(3) = 082/912923
```

Il nostro record logico è dunque formato da due campi che abbiamo chiamato NOM\$(I) e TEL\$(I).

Come su molti altri sistemi, prima di utilizzare un file dobbiamo in qualche modo informare la macchina delle nostre intenzioni; è questo il senso della istruzione di "apertura file" che sul TEXAS ha la seguente sintassi:

OPEN # numero file: nome dispositivo, organizzazione file, tipo file, modo di apertura, tipo record.

Esaminiamo un po' più da vicino i parametri da definire:

**Numero di file:** può essere un intero tra 1 e 255. Serve ad identificare nelle istruzioni del programma il file a cui facciamo riferimento.

**Nome dispositivo:** identifica il registratore che vogliamo usare: può essere "CS1" o "CS2".

**Organizzazione file:** con le cassette può essere solo SEQUENTIAL cioè sequenziale. Questa dichiarazione può essere omessa una volta specificato l'uso di un registratore.

**Modo di apertura:** INPUT se vogliamo leggere dati dal nastro; OUTPUT se invece vogliamo scrivere dati sul nastro. Ricordate che il "CS2" cioè il secondo registratore, non può leggere dati ma solo scriverli. Così se il nome dispositivo è "CS2", il modo di apertura può essere solo OUTPUT.

**Tipo file:** INTERNAL o DISPLAY; questo è un punto importante su cui mi soffermerò in seguito. Se non specificate il "tipo file", il calcolatore lo assume DISPLAY.

**Tipo record:** per le cassette può essere solo FIXED cioè possiamo usare solo record FISICI di lunghezza fissa. Possiamo però scegliere tra record lunghi 64, 128 o 192 caratteri. Quando non si specifica la lunghezza del record, essa viene assunta uguale a 64 caratteri. Non sottovalutate l'importanza di questa scelta: in molti casi può risultare addirittura determinante per l'efficienza dei programmi.

Una volta utilizzato il file, per informare il calcolatore che deve chiudere il "canale di comunicazione" con il dispositivo esterno, è disponibile una semplice istruzione:

CLOSE # numero di file dove "numero di file" è il numero, assegnato nella OPEN, del file che vogliamo chiudere. Naturalmente ci sono ancora le istruzioni più importanti, quelle che ci permettono di leggere e scrivere dati sul nastro.

Esse sono rispettivamente:  
INPUT # numero di file: lista di variabili.

PRINT # numero di file: lista di variabili.

A parte il numero di file, esse hanno una sintassi molto simile alla PRINT ed alla INPUT normalmente usate nei vostri programmi. Il motivo è che il TI99/4A gestisce come file anche la tastiera ed il video ed

assegna loro il numero di file uguale a 0. Infatti le istruzioni:

```
100 PRINT "CIAO CIAO"
```

e

```
100 PRINT # 0 : "CIAO CIAO"
```

oppure

```
110 INPUT A$
```

e

```
110 INPUT # 0 : A$
```

sono perfettamente equivalenti. Tornando al nostro esempio, supposto acquisito precedentemente il contenuto dei vettori NOM\$(I) e TEL\$(I) (cioè il nome ed il numero di telefono dei nostri amici) per registrarlo su nastro potremmo scrivere le seguenti istruzioni:

```
300 OPEN # 1 : "CS1", INTERNAL, OUTPUT, FIXED
```

```
310 FOR I = 1 TO 10
```

```
320 PRINT # 1 : NOM$(I), TEL$(I)
```

```
330 NEXT I
```

```
340 CLOSE # 1
```

E, qualora volessimo rileggerlo:

```
500 OPEN # 2 : "CS1", INTERNAL, INPUT, FIXED
```

```
510 FOR I = 1 TO 10
```

```
520 INPUT # 2 : NOM$(I), TEL$(I)
```

```
530 NEXT I
```

```
540 CLOSE # 2
```

Notate come sia necessaria una perfetta corrispondenza dell'ordine di scrittura con quello di lettura, ciò per evitare che variabili diverse si scambino i contenuti. Sia per i file INTERNAL che per quelli DISPLAY, le variabili da stampare su un record devono essere separate da una virgola, un punto e virgola o due punti. Quando la lista di variabili termina con uno di questi separatori, la successiva istruzione riguardante il file fa riferimento al record precedente.

**E ADESSO CHE ANCHE MIO FIGLIO  
E' ENTRATO NELLA ROSSI & ROSSI, CHI  
MI AIUTERA' A PIANIFICARE IL FUTURO  
DELLA ROSSI, ROSSI & ROSSI?**



# IL PERSONAL COMPUTER IBM IL TUO PICCOLO GRANDE AMICO.

La tua azienda sta crescendo. È un momento di grande soddisfazione, che ti ripaga di tutti gli anni dedicati a costruirla. Ma ti procura nuovi e complessi problemi di pianificazione. Oggi, per fortuna, c'è un amico che ti può aiutare ad affrontare il futuro più serenamente: il Personal Computer IBM. Perché controlla tutti quei lavori che, in un momento di crescita, rischierebbero di occuparti troppo tempo: riceve dati, analizza, calcola, stampa e, grazie alla sua potente

memoria e ai minidischi, ti consente di archiviare un'infinità di informazioni. Vedrai, imparerai a dialogare con lui in poche ore.

Con un amico così, ti sarà più facile formulare preventivi e offerte, senza perdere d'occhio il tuo margine di profitto: vuoi conoscerlo meglio? Rivolgiti ai concessionari IBM. Scegli quello che ti è più comodo, nell'elenco della pagina che segue.



IBM Italia  
Distribuzione Prodotti srl



Il Personal Computer IBM contiene un microprocessore a 16 bit e una memoria di utilizzo che raggiunge i 640 Kbyte, e può essere dotato di un video a colori e di un processore matematico. E, grazie ai dischi fissi, la capacità massima di memoria del sistema è di 21 Mbyte in linea. Inoltre, puoi facilmente collegarti con un altro Personal Computer IBM, con elaboratori più potenti e con la rete dei Centri Servizi Elaborazione Dati della IBM.

**Sistemi operativi:** DOS 1 - DOS 2 - UCSD - CPM-86. **Supporti per le comunicazioni:** Asincrono - SDLC - BSC - Emulazione: 3101-3270. **Linguaggi:** tutti i principali e in più l'APL. **Programmi applicativi per:** aziende e servizi - produttività individuale - ufficio moderno - calcolo tecnico e scientifico - applicazioni professionali - didattica.



## I file su cassetta nel TI99/4A

Nel nostro esempio, ad ogni ciclo del FOR stampavamo o leggevamo le variabili NOM\$(I) e TEL\$(I) in un singolo record. Ma le istruzioni: 320 PRINT # 1 : NOM\$(I), TEL\$(I)

e  
320 PRINT # 1 : NOM\$( I),  
325 PRINT # 1 : TEL\$( I)

oppure  
520 INPUT # 1 : NOM\$( I),  
525 INPUT # 1 : TEL\$( I)

e  
520 INPUT # 1 : NOM\$( I), TEL\$( I)

sono perfettamente equivalenti: siccome la 320 PRINT # 1 : NOM\$( I), e la 520 INPUT # 1 : NOM\$( I), terminano con un separatore, in questo caso la virgola, allora la successiva istruzione fa riferimento allo stesso record.

Abbiamo però che, pur esatte formalmente, le parti di programma che avete appena letto sono quanto di peggio si possa scrivere almeno nel nostro caso. Bisogna infatti tenere presenti alcune piccole particolarità di cui parleremo adesso.

### Internal o display?

Le differenze fondamentali tra i due tipi di file sono:

1) **DIVERSA RAPPRESENTAZIONE DEI DATI.** Quelli di tipo INTERNAL sono scritti con la codifica interna del computer; quelli di tipo DISPLAY sono invece scritti con i normali caratteri ASCII standard.

Questo perchè il TI99/4A gestisce con la tecnica del file quasi tutti i dispositivi esterni collegabili.

Tra i "nomi dispositivo" specificabili c'è infatti anche "RS 232" cioè l'interfaccia seriale che di solito unisce una stampante al computer. Dunque per stampe cartacee è indispensabile il formato display.

2) **DIVERSO FORMATO DI STAMPA.**

Per il tipo *INTERNAL*: I dati numerici sono *sempre* lunghi 9 caratteri. Quelli alfanumerici sono

lunghi quanto la stringa da stampare più uno. Ecco alcuni esempi:

DATO	TIPO	LUNGHEZZA
12	numerico	9
12	alfanumerico	3
PIPPPO	alfanumerico	6
9876543294	numerico	9

Il primo carattere di ogni lista alfanumerica indica la lunghezza del dato stesso.

Attenzione: se il numero totale dei caratteri delle variabili da stampare nel record supera la capacità del record stesso, si ha un messaggio di errore ed il programma si ferma. Dobbiamo quindi valutare in anticipo il numero di caratteri massimo che può raggiungere la nostra lista di variabili da stampare in ogni record. Così ad esempio in un record *fisico* lungo 64 caratteri possiamo registrare al massimo 7 dati numerici.

Per il tipo *DISPLAY*: la lunghezza del dato è uguale al numero di caratteri che lo formano. Esempi:

DATO	TIPO	LUNGHEZZA
12	numerico	3
12	alfanumerico	2
PIPPPO	alfanumerico	5
9876543294	numerico	10

Essi devono però al momento della stampa, essere separati da una virgola che viene inserita nella lista di stampa.

Volendo registrare i record del nostro esempio in formato *DISPLAY*, avremmo dovuto scrivere:  
300 OPEN # 2 : "CS1", DISPLAY, OUTPUT, FIXED

310 FOR I = 1 TO 10

320 PRINT # 2 : NOM\$(I); "," ; TEL\$(I); "," ; "

330 NEXT I

340 CLOSE # 2

e, per rileggerli dovremo semplicemente cambiare le istruzioni 300 e 320 in:

300 OPEN # 2 : "CS1", DISPLAY, INPUT, FIXED

320 INPUT # 2 : NOM\$(I); "," ; TEL\$(I); "," ; "

Se la lunghezza delle variabili da scrivere nel record supera la capacità fissata del record stesso, non si ha un *FILE ERROR*, ma l'ultimo dato diventa il primo del record successivo. Per evitare errori di lettura, tuttavia, è bene che anche in questo caso si valuti attentamente il numero di caratteri delle variabili da stampare.

Nelle normali operazioni di lettura/scrittura da nastro, consiglio senz'altro il formato *INTERNAL*. Ma resta da stabilire ... *quanti caratteri dopo il fixed?*

Abbiamo già definito un record logico: esso è l'unità logica di informazione di un archivio. Abbiamo però parlato anche di *RECORD FISICO*: nel nostro caso è la porzione fisica di nastro su cui vengono registrate le informazioni. Ogni volta che scrive un *RECORD FISICO*, il nostro computer si ferma per qualche attimo (o meglio ferma il registratore). Noi abbiamo la possibilità, attraverso l'istruzione *OPEN*, di stabilire quanti caratteri esso debba stampare per ogni record fisico (64, 128, 192). Tornando al nostro esempio, supposto che i nomi dei nostri amici siano lunghi al massimo 20 caratteri e riservando 10 caratteri per il numero di telefono, il nostro *RECORD LOGICO* è lungo al massimo 32 caratteri (infatti due caratteri indicano la lunghezza delle stringhe *NOM\$(I)* e *TEL\$(I)*).

Per come abbiamo scritto il programma, però, il record *FISICO* è lungo 64 caratteri. Rimangono perciò inutilizzati ben 32 caratteri.

La prima soluzione che ci viene in mente è registrare 2 record logici per ogni record fisico. Ma se consideriamo che la differenza di tempo di registrazione per ogni record tra un *FIXED 64* e un *FIXED 192* è di soli due secondi risulta lampante che la soluzione ottimale è registrare 6 record logici per ogni record fisico lungo 192 caratteri. In solo 8 secondi dunque stamperemo 6 dei nostri record logici contro i 6 secondi ne-

# ECO CHI TI AIUTERA' AD ANDARE D'AMORE E D'ACCORDO CON TUO NUOVO AMICO.



Il tuo concessionario IBM. Ti aiuterà a ottenere il massimo dal tuo Personal Computer IBM. Ti garantirà un'assistenza puntuale e un servizio all'altezza del nome IBM, che in tutto il mondo significa efficienza e affidabilità. Per una lunga e proficua amicizia fra te e il tuo Personal Computer IBM. Per acquisti superiori alle 20 unità puoi anche rivolgerti alle filiali IBM. E per ulteriori informazioni su eventuali punti di vendita che non compaiono sull'elenco, telefona a: 02/21752360 oppure 06/54864962.

## ABRUZZI/MOLISE

Campano - ITALDATA SRL - Via Tiburtina, 75 - Tel. 055.505424  
Pescobasso - PUBLISISTEMI SRL - Via S. Antonio Abate, 236 - Tel. 0874.98444

## BASILICATA

Potenza - I.P.E.S. SPA - Via Sanremo, 79 - Tel. 0971.43293

## CALABRIA

Cosenza - CALIO SRL - Via N. Serra, 90 - Tel. 0984.32807

## CAMPANIA

Cava dei Tirreni - METELLIANA SPA - Via Mandoli, 16  
Napoli - COMPUTER SISTEMI SRL - Via A. De Gasperi, 45  
Tel. 081.312322  
Saierno - OMNIA SRL - C.so Garibaldi, 47 - Tel. 089.220366  
S. Marco Capuovestre - GENERAL SYSTEMS SRL - Via Unità d'Italia, 2/23 - Tel. 0923.81100

## EMILIA

Bologna  
ARCO SAS - Via Bernini, 1 - Tel. 051.393274  
CMB INFORMATICA SRL - Via Arcovegno, 74/10 - Tel. 051.323594  
LUCCA SYSTEMS SRL - Via Farini, 33/A - Tel. 051.231569  
SYSDATA ITALIA SPA - Via Massimo d'Azeglio, 58 - Tel. 051.330021

## Capri

DATA SRL - Via B. Peruzzi, 12 - Tel. 059.688090  
LUNIDATX SPA - Via Biondo, 6 - Tel. 059.689555  
Ferrara - MARKET COMPUTERS SRL - Via Bologna, 84 - Tel. 0532.35867

## Forlì

C.E.D.A.F. COOP ELAB. DATI - Via Zenchini, 57 - Tel. 0543.65402  
I.C.O.T. IMPIANTI SRL - Via Codazzi, 10 - Tel. 0543.723014  
Imola - PALAZZO DONATO - Via Emilia, 23/A - Tel. 0542.20195  
Piacenza - RCM COMPUTER SAS - C.so Vittorio Emanuele II, 96 - Tel. 0523.37388

## Reggio Emilia

A.F.E.D. ELABORAZIONE DATI - Via Filippo Re, 17 - Tel. 0522.28721  
MEMAR ELECTRONIC SRL - Via Melato, 13 - Tel. 0522.94220  
Rimini - HARD & SOFT SYSTEMS SRL - Via Valturio, 43  
Tel. 0541.773443

## LAZIO

Frosinone - SAIU ELETTRONICA SRL - Via Vado del Tufo, 85 - Tel. 0775.83093

## Roma

CERVED SPA - Via Appia Nuova, 696 - Tel. 06.7940241  
DATAOFFICE SPA - Via Sicilia, 205 - Tel. 06.4765468  
ELECTRA 85 SPA - Via G. Valmorana, 83 - Tel. 06.8127324  
GEDIN SRL - L.go D. De Dominicis, 7 - Tel. 06.432183  
I.S.E.D. SPA - Via Tiburtina, 1236 - Tel. 06.4125551  
ISITAL SISTEMI PER L'INFORMATICA SPA - Via S.S. Appolati, 66 - Tel. 06.6793477  
JACOBISSI SPA - Via V. Brancati, 64 - Tel. 06.50091242  
MEMORY COMPUTER SRL - Via Aureliana, 39 - Tel. 06.4758366  
MICROCOMP SPA - Via M. Gelsomini, 28/30 - Tel. 06.577484  
NICA DIFF INF SRL - V.le Parioli, 40 - Tel. 06.872603  
SAPES SRL - V.le Tito Livio, 12 - Tel. 06.3453536  
VAL D'AZEL SRL - P.zza S. Anastasio, 3 - Tel. 06.6789663  
Viterbo - ITALBYTE SRL - V.le Trento - Pal. Garbini - Tel. 0761.221333

## LIGURIA

DIFFER SRL - Via XX Settembre, 31/4 - Tel. 010.586238  
ELABORATION PROCESS SPA - Via Fieschi, 3/6 - Tel. 010.565784  
SIFEX SRL - Via SS. Giacomo e Filippo, 13R - Tel. 010.8734445  
Sanremo - DIFFEL C/O D. RCS SRL - Via Helmsere, 8 - Tel. 0194.72435

## LOMBARDIA

Arbino - C.N.O.A. INFORMATICA SAS - Via Provinciale, 86 Comenduno - Tel. 055.751784

Assago - TRANSDATA SRL - Mi Fiori Pal. E3 Str. 1 - Tel. 02.824940  
Bergamo - SELTERING SPA - Via Verdi, 31 - Tel. 035.248256  
Brescia  
BICO SERVICE SRL - Via Pastrengo, 5 - Tel. 030.590555  
MICROSELT SRL - Via Cipro, 33 - Tel. 030.224246  
SELTERING SPA - Via Cipro, 33 - Tel. 030.220391  
Como - BRUNO SRL - Via Robini, 5 - Tel. 031.260338  
Lecce - ZECCA UFFICIO SPA - Viale Dante, 14 - Tel. 0341.373291  
Lodi - ZUCCHETTI SPA - C.so Mazzini, 39 - Tel. 0371.54827  
Mantova - REPLICA COMPUTER SRL - Via G. Arrivabene, 22  
Tel. 0376.388212

## Milano

AMUFFICIO SPA - Via Desenzano, 7 - Tel. 02.4080275  
B.O.M. SAS - V.le Tunisia, 50 - Tel. 02.6599076  
C.S.A. COMM. SRL - Via Farini, 82 - Tel. 02.6888433  
DATA OPTIMIZATION SRL - Via Massacio, 12 - Tel. 02.4957876  
ECS ITALIA SRL - C.so Monforte, 15 - Tel. 02.780213  
EDELKTRON SRL - C.so Sempione, 39 - Tel. 02.1493603  
ELEDRA 85 SPA - Viale Elvezia, 18 - Tel. 02.349751  
GENERAL ELECTRIC INFORMATION SERVICES SPA - Via Regina Giovanna, 29 - Tel. 02.3870131  
HOMIC PERSONAL COMPUTER SRL - Piazza De Angeli, 3 - Tel. 02.4988201  
HUGNOT LUIGI LUCIANO - Via De Togni, 10 - Tel. 02.873190  
IL NUOVO UFFICIO SISTEMI SNC - Via Priv. del Don, 2 - Tel. 02.8350780  
MICROTECH SRL - Via F.lli Bronzetti, 20 - Tel. 02.736609  
SIRIO SHOP SRL - Viale Certosa, 148 - Tel. 3010051  
SIRIO SRL - Viale Mame, 10 - Tel. 02.749196  
S.D.I. STUDIO DI INFORMATICA SPA - Via G. Winckelmann, 1 - Tel. 02.4223305  
Manza - EDCONSULT SRL - Via Rosmini, 3 - Tel. 039.389850

## Pavia

ITC INFORMATICA SRL - Strada Nuova, 86 - Tel. 0382.303201  
S. Antonio Mantov. - ANTEK COMPUTER SAS - Via Manzoni, 49 - Tel. 0376.398759  
Sondrio - G.P.D. SRL OFF. AUTOM. - V.le N. Sauro, 28 - Tel. 0342.218561

## Varese

ELAMEC SPA - Via Sebenico, 12 - Tel. 0332.264135  
VEGA SPA - Via Silvestro Sanvito, 103 - Tel. 0332.229374  
Vigevano - LOGICA INFORMATICA SRL - Via Montegrappo, 32 - Tel. 0331.81888  
Vimercate - DATA PROGRES SRL - Via E. Emanuele, 44/A - Tel. 039.667423  
Vimodrone - OMEGA DATA SRL - Strada Padana Sup. 317 - Tel. 02.2504121

## MARCHE

Jesi - SYST HOUSE A.P.R.A. SRL - V.le Cavallotti, 9 - Tel. 0533.28743  
Moie - S.E.D.A. SPA - P.zza S. Maria - Tel. 0731.70345  
COMPTON COMPUTER & OFFICE SRL - Via Mazzini, 73 - Tel. 0734.61770

## PIEMONTE

Alessandria - INFORMATICA SERVICE SRL - Via Isenzo, 63 - Tel. 0131.445817  
Ava - HASTA DATI SNC - Via Silvio Morando, 6/A - Tel. 0141.61856

## TORINO

BIELLA  
BIELLA SPA - Via Losana, 9 - Tel. 015.24915  
V.I.P. COMPUTERS SRL - Via Repubblica, 39 - Tel. 015.27106  
Cuneo - I.D.S. INF. DATA SYST SRL - Viale Marullo, 157 - Tel. 0163.25327  
Vercelli - SISTEMI SRL - Via Giolitti, 26 - Tel. 0171.55475  
Genova - EUROSYSTEMS SPA - Bovo S.S. 20/28 - Tel. 0172.68176  
Tortona  
DIFFERISCIPATE VENCO SRL - C.so Matteotti, 32A - Tel. 0151.545225  
PROGRAMMA SPA - C.so Svizzera, 185 - Tel. 011.746421  
SISTEMI SPA - C.so Peschiera, 49 - Tel. 0131.538676

SOFTPEC SRL - C.so San Maurizio, 79 - Tel. 011.8396444  
Vercelli - ANALOG SNC - Via Dionisotti, 18 - Tel. 011.6011605  
PUGLIE  
Bari - PASED SRL - Via Calefati, 134/136 - Tel. 080.481488  
Foggia - MASELLI PER L'UFFICIO - Via L. Zuppetta, 355A  
Tel. 0881.78014  
Lecce - I.P.E.S. SPA - Via Oberdan, 29 - Tel. 0832.33904  
Maggiù - S.V.I.C. SRL - Via V. Emanuele, 121 - Tel. 0836.21004

## SARDIGNIA

Castell. - C.D.S. SAS - Via Sonnino, 108 - Tel. 070.650756

## SICILIA

Catania  
ASIA COMPUTER SRL - Via S. Euplio, 13 - Tel. 095.328844  
COMPUTER SYSTEMS SRL - Via Ruggero di Lauria, 87 - Tel. 095.493777  
Messina - SICILIT FORNITURE SPA - Via Don Blasco, 75 - Tel. 090.292387  
Palermo  
SERCOM ITALIA SRL - Via Scuti, 180 - Tel. 091.261041  
SIPREL SRL - Via Serradifalco, 145 - Tel. 091.573440  
TESI SRL - Via R. Notaroberto, 41 - Tel. 091.260549  
Trapani - TESI SRL - Via Palermo Abate, 2 - Tel. 0923.20026

## TOSCANA

Empoli - SESA DISTRIBUZIONE SRL - Via XI Febbraio, 24/B - Tel. 0571.72148  
Firenze  
DATA COMP SCRL - Via di Novoli, 23/H - Tel. 055.437968  
SESA DISTRIBUZIONE SRL - Lungarno Ferrucci, 19R - Tel. 055.6811602  
Prato - C.C.S. SAS - Viale Repubblica, 298 - Tel. 0574.580222  
Siena - SILEC SRL - Via Sicilia, 5 - Belvedere - Tel. 0577.94065  
Viareggio - DELPHI SRL - Via Aurelia Sud, 39 - Tel. 0584.395161

## TRIVENETO

Bassano D/Grappa - C.P.E. - Piazzetta Poste, 9 - Tel. 0424.20395  
Belluno  
DE PRA SRL - Via I. Caffi, 18 - Tel. 0437.22343  
SFC COMP. SVST. SRL - Via Petre, 32 - Tel. 0437.70826  
Bologna - BOPAM SAS - C. Battisti, 32 - Tel. 0471.30131  
Castelfranco Ven. - EDS SRL - Via S. Pio X, 154 - Tel. 0423.490178

## Padova

CERVED ENGINEERING SPA - C.so Stati Uniti, 14 - Tel. 042.760733  
S.I.C. ITALIA SRL - Via Fontomba, 8 - Tel. 049.45555  
SYSTEM ROS SAS - P.zza De Gasperi, 14 - Tel. 049.38412  
SOGEDA SPA - Via Marsala, 29 - Tel. 049.655886  
S. Donà di Piave - COMPUTIME SRL - Piazzetta Ricci, 63  
Tel. 0421.25448

## Trento

SEDA SPA - Via Sighere, 79 - Tel. 0461.984564  
SICANC - COMPUTER SHOP - Via Prato, 22 - Tel. 0461.29154  
Treviso - INFORMATICA TRE SRL - Via della Repubblica, 19 - Tel. 0422.65993  
Trieste - DITTA MURRI - Via A. Diaz, 24/A - Tel. 040.733253

## Udine

D.E.U. SRL - Via Di Prampero, 3/7 - Tel. 0432.204402

## VAL D'AOSTA

A.S.U. SRL - Via Tavagnacco, 89 - Tel. 0432.482086

## Verona

PRAGMA SOFTWARE SRL - Via Carmelitani Scali, 20 - Tel. 045.596400  
SEVER DI G. SERENI - Via Locatelli, 10 - Tel. 045.31331  
Vicenza - ALFA DATA SRL - Via Milano, 110 - Tel. 0444.31865

## UMBRIA

Perugia - PUFFICCIUCCI SNC - Via XX Settembre, 148/C - Tel. 075.22992

## Ferrari

DIPS SRL - Via Piacinotti, 6 - Tel. 0744.58247

## VALDAOSTA

Aosta  
INFORMATIQUE SAS - Av. Du Cons. De Commis. 16 - Tel. 0165.2242

Per maggiori informazioni, compila e spedisce questo tagliando al tuo concessionario di zona.

Nome		Cognome	
Via		N°	
Cap		Città	

Personal Software 12/13



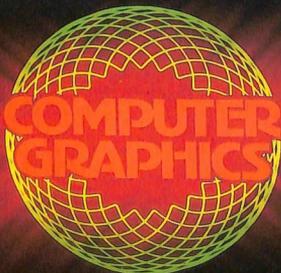


milano 7/10 febbraio 1984

# Evoluzione computer

L'appuntamento annuale con il meglio della produzione americana nel settore dell'informatica: computer, periferiche, sistemi di word processing e trasferimento dati, software ed accessori.

Tutte le case più prestigiose del settore saranno presenti a questa manifestazione che si rivolge ad un pubblico altamente qualificato e desideroso di mantenersi aggiornato sulle ultime novità "made in U.S.A."



In occasione del 20° anniversario del Centro Commerciale Americano in Italia, la XIII edizione di EDP USA dedica un intero padiglione ad una novità assoluta: la prima mostra commerciale di COMPUTER GRAPHICS.

Su questo tema specifico, nei giorni 8 e 9 febbraio, verranno organizzati due seminari: uno "tutorial" per un primo approccio alle tematiche del Computer Graphics ed un altro "tecnico" per illustrare agli specialisti gli sviluppi più recenti del settore.



Per ulteriori informazioni:

**CENTRO COMMERCIALE  
AMERICANO**

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





## I file su cassetta nel TI99/4A

cessari a registrarne uno solo nel nostro primo programma. Ecco come potremmo fare:

200 OPEN # 2 : "CSI", INTERNAL, OUTPUT, FIXED 192  
210 FOR I = 1 TO 12 STEP 6  
220 FOR J = 1 TO I + 4  
230 PRINT # 2 : NOM\$(J), TEL\$(J),

240 NEXT J  
250 PRINT # 2 : NOM\$(J), TEL\$(J)

260 NEXT I  
270 CLOSE # 2  
e per rileggere:

300 OPEN # 2 : "CSI", INTERNAL, INPUT, FIXED 192  
310 FOR I = 1 TO 12 STEP 6  
320 FOR J = 1 TO I + 4  
330 INPUT # 2 : NOM\$(J), TEL\$(J),  
340 NEXT J

350 INPUT # 2 : NOM\$(J), TEL\$(J)

360 NEXT I  
370 CLOSE # 2

Il ciclo di FOR sulla variabile I serve a dividere i due blocchi di registrazione, mentre il FOR sulla J (linee 220, 230, 240) serve a scrivere sullo stesso record FISICO cinque dei nostri record logici ognuno formato da un nome e da un numero di telefono.

Ciò è possibile in quanto l'istruzione 230 termina con una virgola.

L'istruzione 250 stampa il sesto record logico e chiude la stampa nel record fisico precedente perchè non termina con *alcun separatore*. Lo stesso discorso vale per la routine di input.

Il "segreto" dunque per una corretta gestione dei file è nello scegliere

sempre una FIXED 192 ed "impacchettare" il maggior numero di informazioni possibili in 192 caratteri con un accorto uso dei separatori.

### Conclusioni

Anche se ottimizzare programmi che gestiscono file può non sembrare facile, almeno a chi affronta per la prima volta simili problematiche, alla luce di quanto finora detto i punti da considerare non sono molti. Con un po' di pratica e di accuratezza tutto sembrerà più facile e gratificante soprattutto per chi non si accontenta più solo di giochini e cerca di affrontare argomenti più "seri" ma non per questo meno divertenti. ■

# È vero: piccolo è bello!

## Alla scoperta dello ZX SPECTRUM

a cura di Rita Bonelli

ZX Spectrum è l'ultimo nato della famiglia Sinclair. È un calcolatore a colori di piccole dimensioni, ma di grandissime possibilità. Imparare a usarlo bene può essere fonte di molte piacevoli scoperte. Questo libro vi aiuta a raggiungere lo scopo. In 35 brevi e facilissimi capitoli non solo imparerete tutto sulla programmazione in BASIC, ma arriverete anche a usare efficientemente il registratore e a sfruttare al meglio le stampe. Soprattutto capirete la differenza tra il vostro Spectrum e gli altri computer.

320 pagine. Lire 22.000 Codice 337 B

GRUPPO  
EDITORIALE  
JACKSON



SCONTO 20% AGLI ABBONATI  
FINO AL 28-2-84



a cura di  
Rita Bonelli

Per ordinare il volume utilizzare l'apposito tagliando inserito in fondo alla rivista

# Come salvare capra e cavoli con lo ZX Spectrum?

## Tecniche di uso generale per la soluzione automatica di problemi non numerici

di Franco Sardo

**C**he il computer, personal o no, possa essere utilizzato con grande profitto nella soluzione di problemi numerici, è cosa ovvia e risaputa. Come esso possa essere utilizzato invece per risolvere problemi decisionali, è noto ma meno ovvio.

Di fronte alle decisioni importanti tutti noi vorremmo poter contare su un elaboratore, considerato intelligenza superiore secondo una visione fumettistica, che vagli tutte le possibilità di scelta fino alle estreme conseguenze, e, a ragion veduta, ci indichi la strada migliore da percorrere.

Tutto ciò è effettivamente possibile applicando le tecniche dell'intelligenza artificiale.

Dimostrerò, con l'ausilio di tre esempi, che è possibile, ogni volta che il problema consista nella scelta fra un numero limitato di decisioni, e purché i risultati delle decisioni siano definibili e rappresentabili con simboli, che esiste uno schema di programma di uso generale, quasi sempre applicabile salvo piccole modifiche, in grado di risolvere il problema.

Un esempio tipico di programmi decisionali è quello degli Scacchi e

degli altri giochi di scacchiera, come l'Otello e la Dama.

Questi programmi però, pur adottando una tecnica di ricerca su albero analoga a quella che illustrerò più avanti, sono troppo specifici, e quindi di applicazione non generalizzabile.

Lo schema proposto si basa su una ricerca su albero di tipo "depth first", con possibilità di backtrack, cioè di risalita ai nodi precedenti nel caso in cui un ramo si riveli infruttuoso.

Esso prende spunto, sia pur con profonde modifiche, dall'articolo *Il giro del cavallo* di G. Morpurgo, comparso sul n. 8/9 di *Personal Software*.

La tecnica proposta presuppone che ad ogni nodo sia possibile un numero ben definito di scelte, e che ogni scelta sia rappresentata da un numero; il cammino percorso nell'albero delle decisioni sarà quindi rappresentabile da una sequenza di numeri; ad esempio

B\$ = "31524"

significa che, partendo dalla situazione iniziale, si è fatta prima la scelta di tipo 3, poi quella di tipo 1, poi quella di tipo 5, e così via.

Sarà così possibile ad ogni ciclo determinare la situazione in cui ci si è venuti a trovare, applicando alla situazione di partenza la prima scelta, e poi alla nuova situazione via via le varie scelte effettuate.

Dovrà però esistere un algoritmo in grado di modificare la situazione in funzione del tipo di scelta applicata; cioè, nel caso che vedremo più avanti della capra e dei cavoli, la scelta 1 consisterà nel trasferimento della capra, la scelta 2 nel trasferimento dei cavoli e la scelta 3 nel trasferimento del lupo.

Se la situazione viene rappresentata da una stringa A\$, in cui all'ini-

zio i tre elementi si trovano all'estremità di sinistra, per un trasferimento di tipo 1 l'algoritmo dovrà cancellare la capra, (che indichiamo con P come pecora per distinguerne l'iniziale dai cavoli), dall'estremità destra e riscriverla all'estremità sinistra. Quindi inizialmente avremo A\$ = "PC L" e dopo il trasferimento A\$ = "CL P" dove P è la pecora, C i cavoli e L il lupo.

Trovare l'algoritmo che operi su una stringa, su un vettore o su una matrice rappresentando numericamente o simbolicamente il mutare della situazione in funzione delle scelte fatte ad ogni nodo è senz'altro il compito più impegnativo, e quello che si differenzia di più da problema a problema, in quanto è legato ai termini stessi del problema.

Nel caso di un gioco di scacchiera, è chiaro che la situazione può essere convenientemente rappresentata da una matrice quadrata contenente i pezzi. Ogni scelta corrisponde ad un diverso spostamento di un pezzo.

In altri casi, come quello della capra e dei cavoli, la situazione può essere rappresentata dalla disposizione dei caratteri in una stringa alfanumerica.

Sta al programmatore trovare di volta in volta il sistema più conveniente su cui sia più facile operare.

Il programma scende in profondità in ogni ramo, continuando ad aggiungere alla stringa B\$, contenente il percorso, i numeri rappresentativi delle scelte fatte. Se una scelta, trasformata nella situazione corrispondente, non viene ritenuta accettabile secondo le regole del problema, o non conveniente da una funzione euristica o non fruttuosa da una funzione barriera ("bound"), viene scartata e si passa ad esaminare la scelta successiva.

Se invece la scelta viene conside-

**Come salvare capra e cavoli con lo ZX Spectrum?**

rata accettabile, il corrispondente numero viene aggiunto alla stringa B\$; se si arriva ad un punto cieco, in cui non è possibile nessuna scelta, o se si supera un eventuale limite di profondità imposto dal programma, si comincia a risalire lungo l'albero accorciando la stringa B\$; quindi si prova un'altra strada aumentando di uno l'ultimo numero della sequenza.

La parte centrale del programma, che genera la scelta, la prova, ed eventualmente la aggiunge al percorso e verifica se si è arrivati alla meta, è un ciclo, che può essere impostato ricorsivamente o meno. Dal punto di vista del funzionamento non cambia niente: la soluzione ricorsiva è soltanto un tantino più elegante (stack del calcolatore permettendo...).

Ad ogni ciclo viene visualizzata la situazione a cui si è pervenuti. Ciò rallenta il raggiungimento della soluzione, ma permette all'utente di seguire il cammino che il calcolatore sta percorrendo.

Poichè in fondo ci interessa poco sapere se bisogna trasferire prima il lupo o i cavoli, ma ci interessa molto vedere come si fa a deciderlo, ritenendo che il rallentamento conseguente possa essere accettabile.

Il ciclo si ferma quando si raggiunge una soluzione, o quando si esauriscono infruttuosamente tutte le possibilità.

In pratica il programma consiste strutturalmente di 6 fasi principali:

1. la inizializzazione delle variabili e il dimensionamento dei vettori,
2. il ciclo di ricerca su albero,
3. la subroutine che chiamerò di derivazione, cioè quella che trasforma la sequenza di numeri rappresentanti le scelte fatte nella situazione attuale,
4. la subroutine che visualizza la si-

- tuazione attuale,
5. la subroutine di riconoscimento della meta, che può anche essere una sola linea di programma,
6. la subroutine che stabilisce la liceità o meno di una mossa in base alle regole del problema.

A questa struttura essenziale si aggiungono alcune parti accessorie, che possono esserci o non esserci.

Esse consistono nelle funzioni barriera, che accelerano il raggiungimento della soluzione eliminando i rami sicuramente improduttivi (v. il problema delle otto regine) o che fermano la ricerca quando ci si spinge troppo in profondità, raggiungendo livelli sicuramente eccessivi (v. il problema della capra e dei cavoli).

Lo schema di programma proposto ha la seguente struttura:

- Inizializzazioni,
- Inizio del ciclo,
  - eventuale funzione barriera,
  - ricostruzione della situazione in base alla sequenza contenuta in B\$ (fase 3),
  - stampa della situazione (fase 4),
  - esame della accettabilità della mossa (fase 6),
  - riconoscimento della meta (fase 5),
    - se la mossa è lecita, aggiungi a B\$ il numero corrispondente alla scelta fatta,
    - se non è lecita, prova la scelta successiva,
    - se non ci sono scelte possibili, risali al nodo superiore,
- Fine ciclo.
- Subroutine che trasforma la sequenza contenuta in B\$ in una situazione.
- Subroutine che visualizza la situazione.
- Subroutine che esamina la liceità della mossa.
- Subroutine che verifica il rag-

giungimento della meta.  
 Questo schema generale è stato applicato nella realizzazione di tre programmi, per lo ZX Spectrum; essi sono:  
 Il giro del cavallo,  
 Le otto regine,  
 Capra e cavoli.

```

60 REM © Franco Sardo 1983
61 REM #####
91 REM ##### DEL CAVALLO#####
92 REM #####
*
95 CLS
100 INPUT "posizione iniziale":G
G=
110 LET RIGORIN=CODE C$(1)-64
111 LET CODI=VAL C$(2)
112 LET B$=""
113 LET I=0
114 LET J=0
115 LET K=0
116 LET L=0
117 LET M=0
118 LET N=0
119 LET O=0
120 LET P=0
121 FOR I=1 TO LEN B$
122 LET B$(I)=?
123 DO SUB 3000
124 NEXT I
125 LET B$=B$+STRS (I+1)
126 REM B$=B$+STRS (I+1)
127 DO SUB 2000
128 NEXT I
129 LET B$=B$+STRS (I+1)
130 LET B$=B$+STRS (I+1)
131 LET B$=B$+STRS (I+1)
132 LET B$=B$+STRS (I+1)
133 LET B$=B$+STRS (I+1)
134 LET B$=B$+STRS (I+1)
135 LET B$=B$+STRS (I+1)
136 LET B$=B$+STRS (I+1)
137 LET B$=B$+STRS (I+1)
138 LET B$=B$+STRS (I+1)
139 LET B$=B$+STRS (I+1)
140 LET B$=B$+STRS (I+1)
141 LET B$=B$+STRS (I+1)
142 LET B$=B$+STRS (I+1)
143 LET B$=B$+STRS (I+1)
144 LET B$=B$+STRS (I+1)
145 LET B$=B$+STRS (I+1)
146 LET B$=B$+STRS (I+1)
147 LET B$=B$+STRS (I+1)
148 LET B$=B$+STRS (I+1)
149 LET B$=B$+STRS (I+1)
150 LET B$=B$+STRS (I+1)
151 LET B$=B$+STRS (I+1)
152 LET B$=B$+STRS (I+1)
153 LET B$=B$+STRS (I+1)
154 LET B$=B$+STRS (I+1)
155 LET B$=B$+STRS (I+1)
156 LET B$=B$+STRS (I+1)
157 LET B$=B$+STRS (I+1)
158 LET B$=B$+STRS (I+1)
159 LET B$=B$+STRS (I+1)
160 LET B$=B$+STRS (I+1)
161 LET B$=B$+STRS (I+1)
162 LET B$=B$+STRS (I+1)
163 LET B$=B$+STRS (I+1)
164 LET B$=B$+STRS (I+1)
165 LET B$=B$+STRS (I+1)
166 LET B$=B$+STRS (I+1)
167 LET B$=B$+STRS (I+1)
168 LET B$=B$+STRS (I+1)
169 LET B$=B$+STRS (I+1)
170 LET B$=B$+STRS (I+1)
171 LET B$=B$+STRS (I+1)
172 LET B$=B$+STRS (I+1)
173 LET B$=B$+STRS (I+1)
174 LET B$=B$+STRS (I+1)
175 LET B$=B$+STRS (I+1)
176 LET B$=B$+STRS (I+1)
177 LET B$=B$+STRS (I+1)
178 LET B$=B$+STRS (I+1)
179 LET B$=B$+STRS (I+1)
180 LET B$=B$+STRS (I+1)
181 LET B$=B$+STRS (I+1)
182 LET B$=B$+STRS (I+1)
183 LET B$=B$+STRS (I+1)
184 LET B$=B$+STRS (I+1)
185 LET B$=B$+STRS (I+1)
186 LET B$=B$+STRS (I+1)
187 LET B$=B$+STRS (I+1)
188 LET B$=B$+STRS (I+1)
189 LET B$=B$+STRS (I+1)
190 LET B$=B$+STRS (I+1)
191 LET B$=B$+STRS (I+1)
192 LET B$=B$+STRS (I+1)
193 LET B$=B$+STRS (I+1)
194 LET B$=B$+STRS (I+1)
195 LET B$=B$+STRS (I+1)
196 LET B$=B$+STRS (I+1)
197 LET B$=B$+STRS (I+1)
198 LET B$=B$+STRS (I+1)
199 LET B$=B$+STRS (I+1)
200 LET B$=B$+STRS (I+1)
201 LET B$=B$+STRS (I+1)
202 LET B$=B$+STRS (I+1)
203 LET B$=B$+STRS (I+1)
204 LET B$=B$+STRS (I+1)
205 LET B$=B$+STRS (I+1)
206 LET B$=B$+STRS (I+1)
207 LET B$=B$+STRS (I+1)
208 LET B$=B$+STRS (I+1)
209 LET B$=B$+STRS (I+1)
210 LET B$=B$+STRS (I+1)
211 LET B$=B$+STRS (I+1)
212 LET B$=B$+STRS (I+1)
213 LET B$=B$+STRS (I+1)
214 LET B$=B$+STRS (I+1)
215 LET B$=B$+STRS (I+1)
216 LET B$=B$+STRS (I+1)
217 LET B$=B$+STRS (I+1)
218 LET B$=B$+STRS (I+1)
219 LET B$=B$+STRS (I+1)
220 LET B$=B$+STRS (I+1)
221 LET B$=B$+STRS (I+1)
222 LET B$=B$+STRS (I+1)
223 LET B$=B$+STRS (I+1)
224 LET B$=B$+STRS (I+1)
225 LET B$=B$+STRS (I+1)
226 LET B$=B$+STRS (I+1)
227 LET B$=B$+STRS (I+1)
228 LET B$=B$+STRS (I+1)
229 LET B$=B$+STRS (I+1)
230 LET B$=B$+STRS (I+1)
231 LET B$=B$+STRS (I+1)
232 LET B$=B$+STRS (I+1)
233 LET B$=B$+STRS (I+1)
234 LET B$=B$+STRS (I+1)
235 LET B$=B$+STRS (I+1)
236 LET B$=B$+STRS (I+1)
237 LET B$=B$+STRS (I+1)
238 LET B$=B$+STRS (I+1)
239 LET B$=B$+STRS (I+1)
240 LET B$=B$+STRS (I+1)
241 LET B$=B$+STRS (I+1)
242 LET B$=B$+STRS (I+1)
243 LET B$=B$+STRS (I+1)
244 LET B$=B$+STRS (I+1)
245 LET B$=B$+STRS (I+1)
246 LET B$=B$+STRS (I+1)
247 LET B$=B$+STRS (I+1)
248 LET B$=B$+STRS (I+1)
249 LET B$=B$+STRS (I+1)
250 LET B$=B$+STRS (I+1)
251 LET B$=B$+STRS (I+1)
252 LET B$=B$+STRS (I+1)
253 LET B$=B$+STRS (I+1)
254 LET B$=B$+STRS (I+1)
255 LET B$=B$+STRS (I+1)
256 LET B$=B$+STRS (I+1)
257 LET B$=B$+STRS (I+1)
258 LET B$=B$+STRS (I+1)
259 LET B$=B$+STRS (I+1)
260 LET B$=B$+STRS (I+1)
261 LET B$=B$+STRS (I+1)
262 LET B$=B$+STRS (I+1)
263 LET B$=B$+STRS (I+1)
264 LET B$=B$+STRS (I+1)
265 LET B$=B$+STRS (I+1)
266 LET B$=B$+STRS (I+1)
267 LET B$=B$+STRS (I+1)
268 LET B$=B$+STRS (I+1)
269 LET B$=B$+STRS (I+1)
270 LET B$=B$+STRS (I+1)
271 LET B$=B$+STRS (I+1)
272 LET B$=B$+STRS (I+1)
273 LET B$=B$+STRS (I+1)
274 LET B$=B$+STRS (I+1)
275 LET B$=B$+STRS (I+1)
276 LET B$=B$+STRS (I+1)
277 LET B$=B$+STRS (I+1)
278 LET B$=B$+STRS (I+1)
279 LET B$=B$+STRS (I+1)
280 LET B$=B$+STRS (I+1)
281 LET B$=B$+STRS (I+1)
282 LET B$=B$+STRS (I+1)
283 LET B$=B$+STRS (I+1)
284 LET B$=B$+STRS (I+1)
285 LET B$=B$+STRS (I+1)
286 LET B$=B$+STRS (I+1)
287 LET B$=B$+STRS (I+1)
288 LET B$=B$+STRS (I+1)
289 LET B$=B$+STRS (I+1)
290 LET B$=B$+STRS (I+1)
291 LET B$=B$+STRS (I+1)
292 LET B$=B$+STRS (I+1)
293 LET B$=B$+STRS (I+1)
294 LET B$=B$+STRS (I+1)
295 LET B$=B$+STRS (I+1)
296 LET B$=B$+STRS (I+1)
297 LET B$=B$+STRS (I+1)
298 LET B$=B$+STRS (I+1)
299 LET B$=B$+STRS (I+1)
300 LET B$=B$+STRS (I+1)
301 LET B$=B$+STRS (I+1)
302 LET B$=B$+STRS (I+1)
303 LET B$=B$+STRS (I+1)
304 LET B$=B$+STRS (I+1)
305 LET B$=B$+STRS (I+1)
306 LET B$=B$+STRS (I+1)
307 LET B$=B$+STRS (I+1)
308 LET B$=B$+STRS (I+1)
309 LET B$=B$+STRS (I+1)
310 LET B$=B$+STRS (I+1)
311 LET B$=B$+STRS (I+1)
312 LET B$=B$+STRS (I+1)
313 LET B$=B$+STRS (I+1)
314 LET B$=B$+STRS (I+1)
315 LET B$=B$+STRS (I+1)
316 LET B$=B$+STRS (I+1)
317 LET B$=B$+STRS (I+1)
318 LET B$=B$+STRS (I+1)
319 LET B$=B$+STRS (I+1)
320 LET B$=B$+STRS (I+1)
321 LET B$=B$+STRS (I+1)
322 LET B$=B$+STRS (I+1)
323 LET B$=B$+STRS (I+1)
324 LET B$=B$+STRS (I+1)
325 LET B$=B$+STRS (I+1)
326 LET B$=B$+STRS (I+1)
327 LET B$=B$+STRS (I+1)
328 LET B$=B$+STRS (I+1)
329 LET B$=B$+STRS (I+1)
330 LET B$=B$+STRS (I+1)
331 LET B$=B$+STRS (I+1)
332 LET B$=B$+STRS (I+1)
333 LET B$=B$+STRS (I+1)
334 LET B$=B$+STRS (I+1)
335 LET B$=B$+STRS (I+1)
336 LET B$=B$+STRS (I+1)
337 LET B$=B$+STRS (I+1)
338 LET B$=B$+STRS (I+1)
339 LET B$=B$+STRS (I+1)
340 LET B$=B$+STRS (I+1)
341 LET B$=B$+STRS (I+1)
342 LET B$=B$+STRS (I+1)
343 LET B$=B$+STRS (I+1)
344 LET B$=B$+STRS (I+1)
345 LET B$=B$+STRS (I+1)
346 LET B$=B$+STRS (I+1)
347 LET B$=B$+STRS (I+1)
348 LET B$=B$+STRS (I+1)
349 LET B$=B$+STRS (I+1)
350 LET B$=B$+STRS (I+1)
351 LET B$=B$+STRS (I+1)
352 LET B$=B$+STRS (I+1)
353 LET B$=B$+STRS (I+1)
354 LET B$=B$+STRS (I+1)
355 LET B$=B$+STRS (I+1)
356 LET B$=B$+STRS (I+1)
357 LET B$=B$+STRS (I+1)
358 LET B$=B$+STRS (I+1)
359 LET B$=B$+STRS (I+1)
360 LET B$=B$+STRS (I+1)
361 LET B$=B$+STRS (I+1)
362 LET B$=B$+STRS (I+1)
363 LET B$=B$+STRS (I+1)
364 LET B$=B$+STRS (I+1)
365 LET B$=B$+STRS (I+1)
366 LET B$=B$+STRS (I+1)
367 LET B$=B$+STRS (I+1)
368 LET B$=B$+STRS (I+1)
369 LET B$=B$+STRS (I+1)
370 LET B$=B$+STRS (I+1)
371 LET B$=B$+STRS (I+1)
372 LET B$=B$+STRS (I+1)
373 LET B$=B$+STRS (I+1)
374 LET B$=B$+STRS (I+1)
375 LET B$=B$+STRS (I+1)
376 LET B$=B$+STRS (I+1)
377 LET B$=B$+STRS (I+1)
378 LET B$=B$+STRS (I+1)
379 LET B$=B$+STRS (I+1)
380 LET B$=B$+STRS (I+1)
381 LET B$=B$+STRS (I+1)
382 LET B$=B$+STRS (I+1)
383 LET B$=B$+STRS (I+1)
384 LET B$=B$+STRS (I+1)
385 LET B$=B$+STRS (I+1)
386 LET B$=B$+STRS (I+1)
387 LET B$=B$+STRS (I+1)
388 LET B$=B$+STRS (I+1)
389 LET B$=B$+STRS (I+1)
390 LET B$=B$+STRS (I+1)
391 LET B$=B$+STRS (I+1)
392 LET B$=B$+STRS (I+1)
393 LET B$=B$+STRS (I+1)
394 LET B$=B$+STRS (I+1)
395 LET B$=B$+STRS (I+1)
396 LET B$=B$+STRS (I+1)
397 LET B$=B$+STRS (I+1)
398 LET B$=B$+STRS (I+1)
399 LET B$=B$+STRS (I+1)
400 LET B$=B$+STRS (I+1)
401 LET B$=B$+STRS (I+1)
402 LET B$=B$+STRS (I+1)
403 LET B$=B$+STRS (I+1)
404 LET B$=B$+STRS (I+1)
405 LET B$=B$+STRS (I+1)
406 LET B$=B$+STRS (I+1)
407 LET B$=B$+STRS (I+1)
408 LET B$=B$+STRS (I+1)
409 LET B$=B$+STRS (I+1)
410 LET B$=B$+STRS (I+1)
411 LET B$=B$+STRS (I+1)
412 LET B$=B$+STRS (I+1)
413 LET B$=B$+STRS (I+1)
414 LET B$=B$+STRS (I+1)
415 LET B$=B$+STRS (I+1)
416 LET B$=B$+STRS (I+1)
417 LET B$=B$+STRS (I+1)
418 LET B$=B$+STRS (I+1)
419 LET B$=B$+STRS (I+1)
420 LET B$=B$+STRS (I+1)
421 LET B$=B$+STRS (I+1)
422 LET B$=B$+STRS (I+1)
423 LET B$=B$+STRS (I+1)
424 LET B$=B$+STRS (I+1)
425 LET B$=B$+STRS (I+1)
426 LET B$=B$+STRS (I+1)
427 LET B$=B$+STRS (I+1)
428 LET B$=B$+STRS (I+1)
429 LET B$=B$+STRS (I+1)
430 LET B$=B$+STRS (I+1)
431 LET B$=B$+STRS (I+1)
432 LET B$=B$+STRS (I+1)
433 LET B$=B$+STRS (I+1)
434 LET B$=B$+STRS (I+1)
435 LET B$=B$+STRS (I+1)
436 LET B$=B$+STRS (I+1)
437 LET B$=B$+STRS (I+1)
438 LET B$=B$+STRS (I+1)
439 LET B$=B$+STRS (I+1)
440 LET B$=B$+STRS (I+1)
441 LET B$=B$+STRS (I+1)
442 LET B$=B$+STRS (I+1)
443 LET B$=B$+STRS (I+1)
444 LET B$=B$+STRS (I+1)
445 LET B$=B$+STRS (I+1)
446 LET B$=B$+STRS (I+1)
447 LET B$=B$+STRS (I+1)
448 LET B$=B$+STRS (I+1)
449 LET B$=B$+STRS (I+1)
450 LET B$=B$+STRS (I+1)
451 LET B$=B$+STRS (I+1)
452 LET B$=B$+STRS (I+1)
453 LET B$=B$+STRS (I+1)
454 LET B$=B$+STRS (I+1)
455 LET B$=B$+STRS (I+1)
456 LET B$=B$+STRS (I+1)
457 LET B$=B$+STRS (I+1)
458 LET B$=B$+STRS (I+1)
459 LET B$=B$+STRS (I+1)
460 LET B$=B$+STRS (I+1)
461 LET B$=B$+STRS (I+1)
462 LET B$=B$+STRS (I+1)
463 LET B$=B$+STRS (I+1)
464 LET B$=B$+STRS (I+1)
465 LET B$=B$+STRS (I+1)
466 LET B$=B$+STRS (I+1)
467 LET B$=B$+STRS (I+1)
468 LET B$=B$+STRS (I+1)
469 LET B$=B$+STRS (I+1)
470 LET B$=B$+STRS (I+1)
471 LET B$=B$+STRS (I+1)
472 LET B$=B$+STRS (I+1)
473 LET B$=B$+STRS (I+1)
474 LET B$=B$+STRS (I+1)
475 LET B$=B$+STRS (I+1)
476 LET B$=B$+STRS (I+1)
477 LET B$=B$+STRS (I+1)
478 LET B$=B$+STRS (I+1)
479 LET B$=B$+STRS (I+1)
480 LET B$=B$+STRS (I+1)
481 LET B$=B$+STRS (I+1)
482 LET B$=B$+STRS (I+1)
483 LET B$=B$+STRS (I+1)
484 LET B$=B$+STRS (I+1)
485 LET B$=B$+STRS (I+1)
486 LET B$=B$+STRS (I+1)
487 LET B$=B$+STRS (I+1)
488 LET B$=B$+STRS (I+1)
489 LET B$=B$+STRS (I+1)
490 LET B$=B$+STRS (I+1)
491 LET B$=B$+STRS (I+1)
492 LET B$=B$+STRS (I+1)
493 LET B$=B$+STRS (I+1)
494 LET B$=B$+STRS (I+1)
495 LET B$=B$+STRS (I+1)
496 LET B$=B$+STRS (I+1)
497 LET B$=B$+STRS (I+1)
498 LET B$=B$+STRS (I+1)
499 LET B$=B$+STRS (I+1)
500 LET B$=B$+STRS (I+1)
501 LET B$=B$+STRS (I+1)
502 LET B$=B$+STRS (I+1)
503 LET B$=B$+STRS (I+1)
504 LET B$=B$+STRS (I+1)
505 LET B$=B$+STRS (I+1)
506 LET B$=B$+STRS (I+1)
507 LET B$=B$+STRS (I+1)
508 LET B$=B$+STRS (I+1)
509 LET B$=B$+STRS (I+1)
510 LET B$=B$+STRS (I+1)
511 LET B$=B$+STRS (I+1)
512 LET B$=B$+STRS (I+1)
513 LET B$=B$+STRS (I+1)
514 LET B$=B$+STRS (I+1)
515 LET B$=B$+STRS (I+1)
516 LET B$=B$+STRS (I+1)
517 LET B$=B$+STRS (I+1)
518 LET B$=B$+STRS (I+1)
519 LET B$=B$+STRS (I+1)
520 LET B$=B$+STRS (I+1)
521 LET B$=B$+STRS (I+1)
522 LET B$=B$+STRS (I+1)
523 LET B$=B$+STRS (I+1)
524 LET B$=B$+STRS (I+1)
525 LET B$=B$+STRS (I+1)
526 LET B$=B$+STRS (I+1)
527 LET B$=B$+STRS (I+1)
528 LET B$=B$+STRS (I+1)
529 LET B$=B$+STRS (I+1)
530 LET B$=B$+STRS (I+1)
531 LET B$=B$+STRS (I+1)
532 LET B$=B$+STRS (I+1)
533 LET B$=B$+STRS (I+1)
534 LET B$=B$+STRS (I+1)
535 LET B$=B$+STRS (I+1)
536 LET B$=B$+STRS (I+1)
537 LET B$=B$+STRS (I+1)
538 LET B$=B$+STRS (I+1)
539 LET B$=B$+STRS (I+1)
540 LET B$=B$+STRS (I+1)
541 LET B$=B$+STRS (I+1)
542 LET B$=B$+STRS (I+1)
543 LET B$=B$+STRS (I+1)
544 LET B$=B$+STRS (I+1)
545 LET B$=B$+STRS (I+1)
546 LET B$=B$+STRS (I+1)
547 LET B$=B$+STRS (I+1)
548 LET B$=B$+STRS (I+1)
549 LET B$=B$+STRS (I+1)
550 LET B$=B$+STRS (I+1)
551 LET B$=B$+STRS (I+1)
552 LET B$=B$+STRS (I+1)
553 LET B$=B$+STRS (I+1)
554 LET B$=B$+STRS (I+1)
555 LET B$=B$+STRS (I+1)
556 LET B$=B$+STRS (I+1)
557 LET B$=B$+STRS (I+1)
558 LET B$=B$+STRS (I+1)
559 LET B$=B$+STRS (I+1)
560 LET B$=B$+STRS (I+1)
561 LET B$=B$+STRS (I+1)
562 LET B$=B$+STRS (I+1)
563 LET B$=B$+STRS (I+1)
564 LET B$=B$+STRS (I+1)
565 LET B$=B$+STRS (I+1)
566 LET B$=B$+STRS (I+1)
567 LET B$=B$+STRS (I+1)
568 LET B$=B$+STRS (I+1)
569 LET B$=B$+STRS (I+1)
570 LET B$=B$+STRS (I+1)
571 LET B$=B$+STRS (I+1)
572 LET B$=B$+STRS (I+1)
573 LET B$=B$+STRS (I+1)
574 LET B$=B$+STRS (I+1)
575 LET B$=B$+STRS (I+1)
576 LET B$=B$+STRS (I+1)
577 LET B$=B$+STRS (I+1)
578 LET B$=B$+STRS (I+1)
579 LET B$=B$+STRS (I+1)
580 LET B$=B$+STRS (I+1)
581 LET B$=B$+STRS (I+1)
582 LET B$=B$+STRS (I+1)
583 LET B$=B$+STRS (I+1)
584 LET B$=B$+STRS (I+1)
585 LET B$=B$+STRS (I+1)
586 LET B$=B$+STRS (I+1)
587 LET B$=B$+STRS (I+1)
588 LET B$=B$+STRS (I+1)
589 LET B$=B$+STRS (I+1)
590 LET B$=B$+STRS (I+1)
591 LET B$=B$+STRS (I+1)
592 LET B$=B$+STRS (I+1)
593 LET B$=B$+STRS (I+1)
594 LET B$=B$+STRS (I+1)
595 LET B$=B$+STRS (I+1)
596 LET B$=B$+STRS (I+1)
597 LET B$=B$+STRS (I+1)
598 LET B$=B$+STRS (I+1)
599 LET B$=B$+STRS (I+1)
600 LET B$=B$+STRS (I+1)
601 LET B$=B$+STRS (I+1)
602 LET B$=B$+STRS (I+1)
603 LET B$=B$+STRS (I+1)
604 LET B$=B$+STRS (I+1)
605 LET B$=B$+STRS (I+1)
606 LET B$=B$+STRS (I+1)
607 LET B$=B$+STRS (I+1)
608 LET B$=B$+STRS (I+1)
609 LET B$=B$+STRS (I+1)
610 LET B$=B$+STRS (I+1)
611 LET B$=B$+STRS (I+1)
612 LET B$=B$+STRS (I+1)
613 LET B$=B$+STRS (I+1)
614 LET B$=B$+STRS (I+1)
615 LET B$=B$+STRS (I+1)
616 LET B$=B$+STRS (I+1)
617 LET B$=B$+STRS (I+1)
618 LET B$=B$+STRS (I+1)
619 LET B$=B$+STRS (I+1)
620 LET B$=B$+STRS (I+1)
621 LET B$=B$+STRS (I+1)
622 LET B$=B$+STRS (I+1)
623 LET B$=B$+STRS (I+1)
624 LET B$=B$+STRS (I+1)
625 LET B$=B$+STRS (I+1)
626 LET B$=B$+STRS (I+1)
627 LET B$=B$+STRS (I+1)
628 LET B$=B$+STRS (I+1)
629 LET B$=B$+STRS (I+1)
630 LET B$=B$+STRS (I+1)
631 LET B$=B$+STRS (I+1)
632 LET B$=B$+STRS (I+1)
633 LET B$=B$+STRS (I+1)
634 LET B$=B$+STRS (I+1)
635 LET B$=B$+STRS (I+1)
636 LET B$=B$+STRS (I+1)
637 LET B$=B$+STRS (I+1)
638 LET B$=B$+STRS (I+1)
639 LET B$=B$+STRS (I+1)
640 LET B$=B$+STRS (I+1)
641 LET B$=B$+STRS (I+1)
642 LET B$=B$+STRS (I+1)
643 LET B$=B$+STRS (I+1)
644 LET B$=B$+STRS (I+1)
645 LET B$=B$+STRS (I+1)
646 LET B$=B$+STRS (I+1)
647 LET B$=B$+STRS (I+1)
648 LET B$=B$+STRS (I+1)
649 LET B$=B$+STRS (I+1)
650 LET B$=B$+STRS (I+1)
651 LET B$=B$+STRS (I+1)
652 LET B$=B$+STRS (I+1)
653 LET B$=B$+STRS (I+1)
654 LET B$=B$+STRS (I+1)
655 LET B$=B$+STRS (I+1)
656 LET B$=B$+STRS (I+1)
657 LET B$=B$+STRS (I+1)
658 LET B$=B$+STRS (I+1)
659 LET B$=B$+STRS (I+1)
660 LET B$=B$+STRS (I+1)
661 LET B$=B$+STRS (I+1)
662 LET B$=B$+STRS (I+1)
663 LET B$=B$+STRS (I+1)
664 LET B$=B$+STRS (I+1)
665 LET B$=B$+STRS (I+1)
666 LET B$=B$+STRS (I+1)
667 LET B$=B$+STRS (I+1)
668 LET B$=B$+STRS (I+1)
669 LET B$=B$+STRS (I+1)
670 LET B$=B$+STRS (I+1)
671 LET B$=B$+STRS (I+1)
672 LET B$=B$+STRS (I+1)
673 LET B$=B$+STRS (I+1)
674 LET B$=B$+STRS (I+1)
675 LET B$=B$+STRS (I+1)
676 LET B$=B$+STRS (I+1)
677 LET B$=B$+STRS (I+1)
678 LET B$=B$+STRS (I+1)
679 LET B$=B$+STRS (I+1)
680 LET B$=B$+STRS (I+1)
681 LET B$=B$+STRS (I+1)
682 LET B$=B$+STRS (I+1)
683 LET B$=B$+STRS (I+1)
684 LET B$=B$+STRS (I+1)
685 LET B$=B$+STRS (I+1)
686 LET B$=B$+STRS (I+1)
687 LET B$=B$+STRS (I+1)
688 LET B$=B$+STRS (I+1)
689 LET B$=B$+STRS (I+1)
690 LET B$=B$+STRS (I+1)
691 LET B$=B$+STRS (I+1)
692 LET B$=B$+STRS (I+1)
693 LET B$=B$+STRS (I+1)
694 LET B$=B$+STRS (I+1)
695 LET B$=B$+STRS (I+1)
696 LET B$=B$+STRS (I+1)
697 LET B$=B$+STRS (I+1)
698 LET B$=B$+STRS (I+1)
699 LET B$=B$+STRS (I+1)
700 LET B$=B$+STRS (I+1)
701 LET B$=B$+STRS (I+1)
702 LET B$=B$+STRS (I+1)
703 LET B$=B$+STRS (I+1)
704 LET B$=B$+STRS (I+1)
705 LET B$=B$+STRS (I+1)
706 LET B$=B$+STRS (I+1)
707 LET B$=B$+STRS (I+1)
708 LET B$=B$+STRS (I+1)
709 LET B$=B$+STRS (I+1)
710 LET B$=B$+STRS (I+1)
711 LET B$=B$+STRS (I+1)
712 LET B$=B$+STRS (I+1)
713 LET B$=B$+STRS (I+1)
714 LET B$=B$+STRS (I+1)
715 LET B$=B$+STRS (I+1)
716 LET B$=B$+STRS (I+1)
717 LET B$=B$+STRS (I+1)
718 LET B$=B$+STRS (I+1
```





## Come salvare capra e cavoli con lo ZX Spectrum?

ne viene riconosciuto dalla lunghezza della stringa B\$, che raggiunge gli otto elementi.

### Capra e cavoli

Il classico problema di salvare capra e cavoli viene risolto utilizzando la stessa tecnica dei programmi precedenti. La situazione viene però rappresentata da una stringa A\$ che all'inizio contiene:

"PCL "

(la capra diventa qui una pecora per distinguere l'iniziale dai cavoli) la funzione di derivazione sposta dall'altra parte del fiume (nel ns. caso, dall'altra parte della stringa) l'elemento corrispondente alla scelta fatta.

Il problema si complica nella sub-

routine che esamina la liceità della mossa, poichè la convivenza dalla stessa parte del fiume della capra e dei cavoli, o del lupo e della capra è legata alla presenza del pastore. Ecco che allora essa viene segnalata dalla variabile M che diventa alternativamente 1 e 0 ad ogni passaggio successivo, rappresentando così la presenza del pastore su una sponda o sull'altra del fiume.

È facile riconoscere in questa tecnica quella dei nodi di minimo o di massimo usata negli scacchi.

Qui il raggiungimento della meta viene riconosciuto dalla presenza dei tre elementi all'estremità destra della stringa.

Come funzione barriera, si è usata quella che impedisce di esplorare nodi oltre il decimo livello. Senza di essa, il programma andrebbe a level-

li sempre più profondi continuando ad effettuare spostamenti improduttivi.

Anche in questo caso, ad ogni soluzione trovata, il programma si ferma: per ripartire occorre dare GO-TO 200.

### Conclusioni

Come si vede, la tecnica proposta è di uso generale, cioè applicabile a problemi anche molto diversi fra loro.

Per essere usata, però, richiede che il problema sia riconducibile ad una struttura ad albero, e che si riesca a determinare una funzione di derivazione, che faccia corrispondere ad ogni tipo di scelta un mutamento della situazione.

# C64

commodore



## a casa vostra subito!

Se volete riceverlo velocemente compilate e spedite in busta il "Coupon CBM 64"

### EXELCO

Via G. Verdi, 23/25  
20095 - CUSANO MILANINO (MILANO)

Descrizione	Qt.	Prezzo unitario	Totale L.
CBM 64 Personal Computer			
Registatore C2N - VC 1530		L. 620.000	
Introduzione basic CBM 64			
Cartuccia Videogioco			
Floppy Disk VC 1541		L. 585.000	
Stampante SEIKOSHA - GP100 VC		L. 550.000	
Reference Guide CBM 64		L. 24.500	
Interfaccia IEEE 488		L. 170.000	

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

Nome

Cognome

Via

Città

Data    C.A.P.

Desidero ricevere la fattura  SI  NO

Partita I.V.A. o, per i privati

Codice Fiscale

Acconto L.

Sarà data precedenza alle spedizioni, se assieme all'ordine verrà incluso un anticipo di almeno L. 10.000. Aggiungere L. 5.000 per il recapito a domicilio. I prezzi vanno maggiorati dell'I.V.A. 18%



ATOM-ACORN

## Controllo della posizione del cursore

Italo A. Albanese

Dal manuale inglese (pag. 16) si apprende che il carattere in stampa è contenuto nella locazione di memoria  $\#E1$ ; non è scritto però che la sua posizione è determinata da tre byte vicini e precisamente:

- La colonna da  $\#E0$ ;
- L'indirizzo del primo dei 32 byte che formano la riga di stampa, da  $\#DE$  e da  $\#DF$  (il byte più significativo in  $\#DF$ ).

Il numero della riga è quindi:

$$((\#DF)\&\#FFFF-\#8000)/32$$

La locazione di memoria alla quale sta puntando il cursore è:

$$((\#DF)\&\#FFFF)(\#E0)$$

Si possono modificare questi byte per spostare il cursore, bisogna però tenere presente quanto segue:

- 1) Il cursore non viene cancellato dalla posizione che occupava prima dello spostamento; ciò può essere ottenuto con  $\#E1 = 0$  oppure CLEAR 0.
- 2) Quando si modifica la colonna del cursore tramite  $\#E0$  ogni nuova riga inizia dalla colonna che abbiamo impostato in questo byte.

Per esempio, se eseguiamo:

$$\#E0 = 10$$

il cursore si posizionerà sulla 10° colonna e la prima riga inizierà da quel punto; arrivati alla 32° colonna il cursore tornerà alla prima colonna della stessa riga e passerà alla riga sottostante solo dopo aver raggiunto la 10° colonna.

- 3)  $\#E0$  può assumere anche valori superiori a 31; il cursore si sposterà di  $(\#E0)/32$  righe alla  $(\#E0)/32$ -esima colonna.
- Se però si supera 127 spariscono cursore e testo.
- 4) L'indirizzo allocato in  $\#DF$ ,  $\#DE$  deve essere un multiplo di 32; altrimenti si ottengono dei comportamenti anomali utilizzabili in circostanze particolari.

Per esempio, se  $\#DF$  è minore di  $\#81$ , il cursore effettuerà lo spostamento previsto e tutto funzionerà normalmente, ma dopo aver completato la nona riga si avrà un ritorno all'inizio dello schermo.

Se invece  $\#DF$  è maggiore di  $\#81$  l'anomalia riguarderà l'ultima riga. Succederà che, arrivati alla

fine dello schermo, anziché tornare all'inizio dell'ultima riga per completarla, la scrittura proseguirà "fuori quadro" e gli ultimi caratteri andranno persi, poi, quando si va a capo tutto lo schermo scivolerà verso l'alto e nell'ultima riga, a sinistra del cursore, si ritroveranno gli stessi caratteri che nella riga precedente occupavano quella posizione.

Questi potranno essere modificati una volta per tutte dopo aver spostato il cursore con i comandi di editing.

- 5) Tutti i byte di posizione sono resettati ai loro valori abituali da HOME e da FORMFEED.
- 6) Dopo un RETURN  $\#E0$  è sempre uguale a zero.
- 7) Uno degli usi più interessanti di questi byte è quello di indirizzare il cursore in un'area di memoria diversa da quella ordinaria.

Per esempio eseguendo  $\#DF = \#82$  e provando a scrivere qualcosa non si vedrà comparire nulla sullo schermo.

Potrà essere reso visibile in seguito, eseguendo:

```
FOR N = #8000 TO #8200 STEP 4 ;!N = #200!  
N;NEXT
```

Ossia copiando nell'area di memoria del VDU il contenuto di quella dove avevamo scritto in precedenza.

Molto semplice, infine, il metodo per creare una "finestra" di testo in un disegno ad alta risoluzione. Avrete certamente notato che, provando a scrivere qualcosa in queste condizioni, compaiono degli strani simboli sulle prime righe dello schermo.

Questi non sono altro che i codici ASCII, rappresentati in binario, dei caratteri che abbiamo scritto.

Per ottenere i caratteri stampati normalmente occorre decodificare il codice ASCII, trasformando le prime righe da righe di disegno ad alta risoluzione a righe di testo a bassa risoluzione.

Per cambiare la risoluzione dello schermo si adopera comunemente il comando CLEAR, qui chiaramente inutilizzabile.

Per sostituirlo si dovrà intervenire direttamente sul byte di controllo del VDU,  $\#B00$ .

Il manuale riporta a pagina 88 (edizione inglese) i valori che devono essere dati a questo byte per ottenere le varie risoluzioni. Essi sono, per le risoluzioni 0, 1a, 1, ..., 4a, 4, rispettivamente:  $\#00$ ,  $\#10$ ,  $\#30$ , ...,  $\#D0$ ,  $\#F0$ .

Ormai il procedimento dovrebbe essere chiaro; quando il VDU inizierà la scansione dello schermo,  $\#B00$  sarà posto a zero.

Dopo un certo periodo, sufficiente per completare le prime righe di testo, il byte di controllo sarà caricato nuovamente con il valore che aveva in precedenza e tutto il ciclo sarà ripetuto all'infinito.

PERSONAL  
SOFTWARE

## Controllo della posizione del cursore

Trovare l'esatto inizio della scansione non pone problemi; basterà fare ricorso all'istruzione WAIT all'inizio di ogni ciclo.

Ecco un esempio di quanto descritto:

```
9990 A = #B000;B = ?A  
9995 DO ?A = 0; ; ?A = B;WAIT;UNTIL 0
```

I due “;” separati da uno spazio della riga 9995 servono a generare un ritardo sufficiente per due righe di testo.

Questo ritardo, a causa della tolleranza dei quarzi che controllano le frequenze del clock e dei sincronismi, dovrà essere aggiunto per ciascun Atom, variando il numero degli spazi.

Se si aggiunge un ulteriore ritardo dopo il WAIT si sposterà la “finestra” verso il basso.

Allo stesso modo dovranno essere spostate le righe di testo da espandere, ad esempio spostando il cursore come ho spiegato in precedenza. ■

## La Texas rassicura gli utenti “home” sul TI-99

La notizia, appena comparsa sulla stampa d'oltreoceano, ha rapidamente fatto il giro del globo, suscitando comprensibile allarme tra gli utilizzatori del popolare TI-99/4A.

Il ritiro della Texas dal mercato degli home computer faceva oltretutto il paio con altre dello stesso tenore (e più drammatiche, si pensi alla quasi-chiusura della Osborne...) ribadendo l'alto grado di competitività in questo settore caratterizzato da sovrappollamento e da guerre dei prezzi feroci. Molto opportunamente adesso giunge una precisazione rassicurante, perlomeno per il discreto parco dei possessori di un TI-99. Comunque, almeno per l'Europa, vengono fornite notizie tranquillizzanti anche per i distributori e per i futuri clienti. Ma lasciamo parlare Irfan Selim, marketing manager europeo per il TI-99/4A: “Circa la futura disponibilità di software e il servizio assistenza, per il primo, valutando i nostri attuali inventari, stimiamo di avere sufficiente materiale per far fronte alle future necessità del mercato sia per quanto riguarda i titoli esistenti sia per quelli recentemente annunciati; quanto al servizio assistenza è assicurato anche oltre il periodo di garanzia”.

Negli Stati Uniti è stato annunciato un ribasso di prezzo ai dettaglianti, che permetterà di offrire il prodotto, al pubblico, ad un prezzo conveniente. In Europa si è provveduto ad un'azione simile che, per l'Italia, ha portato il prezzo al pubblico del TI-99/4A a L. 220.000 + IVA al 18% (fino ad esaurimento degli stock, si presume). Importanti riduzioni di prezzo si avranno anche sul software.

Altre importanti precisazioni: l'uscita dal mercato dell'informatica domestica non riguarda in alcun modo i professional computer che continueranno ad essere forniti dalla divisione Data Systems, e viene anche sottolineato che Texas Instruments intende fermamente rimanere nel mercato delle calcolatrici e in quello dei computer portatili-compatti.

## 1983 : l'inizio

Una grande impresa editoriale  
comincia oggi nella vostra edicola

### Il micro-millennio è cominciato.

Siamo nell'era dell'elettronica e dell'informatica.

Una rivoluzione silenziosa sta cambiando il nostro modo di vivere, pensare, esprimerci.

Una scelta ci sta oggi davanti: subire le novità che ci attendono oppure viverle da protagonisti; impadronirci del futuro o farcene travolgere. Decidiamo!

Varcare le soglie del micro-millennio conoscendone tutti i segreti e oggi possibile.

Oggi c'è E.I. l'enciclopedia dell'elettronica e dell'informatica.

Un'opera unica al mondo, scritta da

specialisti per uomini-protagonisti.

E completa, rigorosa, documentata, facile da capire... anche se parla di elettrotecnica, elettronica allo stato solido, elettronica digitale, microprocessori, comunicazioni, informatica di base, informatica e società.

Tutto quello che volete e dovete sapere sul micro-millennio che ci sta aspettando.



### Enciclopedia di Elettronica e Informatica

#### 50 fascicoli settimanali

- 12 pagine di elettronica digitale e microprocessori
- 16 pagine di informatica (oppure elettronica di base e comunicazioni)
- 1 scheda (2 pagine) di elettrotecnica per ottenere in meno di un anno
- 7 grandi volumi
- 1400 pagine complessive
- 1 volume schede di elettrotecnica

L'opera è arricchita da circa 700 foto e 2200 illustrazioni a colori.

Ogni settimana l'elettronica, l'informatica, l'elettrotecnica in un unico fascicolo.



### Enciclopedia di Elettronica e Informatica Oggi in edicola... domani nella vostra biblioteca



GRUPPO EDITORIALE  
JACKSON

In collaborazione con il Learning Center

TEXAS INSTRUMENTS 

**COMMODORE PET/CBM**

**Istruzione ON ERROR GOTO nn per il VIC 20 e C 64**

Alessandro Guida

Un problema che si fa sempre sentire quando ci si accinge a scrivere programmi di una certa importanza è quello di fare in maniera che gli errori, che possono insorgere durante l'esecuzione, non blocchino il programma stesso. Un esempio classico è dato dalle periferiche spente che causano il famoso DEVICE NOT PRESENT. Da qui nasce la necessità di potere intercettare qualsiasi errore e di conoscerne la natura e la linea in cui si è verificato.

La routine che ora vedremo ci dà queste possibilità. In particolare al verificarsi di un errore l'esecuzione del programma, anziché arrestarsi, prosegue ad una linea il cui numero, che possiamo scegliere a piacere, è contenuto nella variabile LN%.

Se la linea non esiste si ha l'errore UNDEF'D STATEMENT.

Inoltre la variabile ST conterrà il codice dell'errore e in LE% potremo leggere il numero della linea in cui questo si è verificato.

Poiché la routine è in linguaggio macchina è riportata sotto forma di codici decimali nei listati 2 e 3. Il primo è per il VIC 20 e il secondo per il Commodore 64.

Nel listato 1 è riportato un esempio applicativo.

LA ROUTINE L.M. Quando l'interprete BASIC incontra un errore, di qualsiasi natura, salta ad una routine (ERROR ROUTINE) il cui indirizzo di partenza è conservato nel vettore \$0300, \$0301 (768, 769). Questa routine è proprio quella che termina con la stampa del messaggio e con il READY.

Il contenuto standard del vettore è \$A43A (naturalmente nell'ordine LB,HB) ma poiché risiede in RAM è possibile cambiarlo con l'indirizzo della nostra routine, che è mostrata disassemblata nel listato 4.

L'unica differenza tra la routine del VIC 20 e quella del 64 risiede negli indirizzi delle subroutine. Bisogna notare che la nuova routine controlla il contenuto della locazione \$3A, e se lo trova uguale a 255 ritorna alla ERROR ROUTINE originale. Questo perché in \$39, \$3A è sempre conservato il numero della linea BASIC corrente.

Se, però, una linea viene eseguita in modo diretto alla \$3A contiene 255. In questa maniera gli errori commessi da tastiera vengono normalmente segnala-

ti da un messaggio standard, mentre quelli incontrati durante l'esecuzione di un programma vengono intercettati.

VIC20	C64	DESCRIZIONE
\$90 \$39, 3A \$45, 46 \$47, 48 \$7A, 7B	\$90 \$39, 3A \$45, 46 \$47, 48 \$7A, 7B	Contiene il valore di ST. Num. linea BAS. corrente. Nome variabile corrente. Indirizzo variabile corrente. Puntatore BASIC.
D0EB	\$B0EB	Cerca la variabile definita in \$45, 46 e ne mette l'indirizzo in \$47, 48. Se la var. non c'è la crea.
\$C617	\$A617	Cerca la linea di programma avente il numero contenuto in \$14, 15, e ne mette l'indirizzo in \$5F, 60.
\$C43A	\$A43A	ERROR ROUTINE originale. il registro X deve contenere il numero dell'errore da stampare.

Riquadro 1. Routine e locazioni di memoria utilizzate.

1	Too many file open
2	File open
3	File not open
4	File not found
5	Device not present
6	Not input file
7	Not output file
8	Missing file name
9	Illegal device
10	Next without for
11	Syntax
12	Return without gosub
13	Out of data
14	Illegal quantity
15	Overflow
16	Out of memory
17	Undef'd statement
18	Bad subscript
19	Redim'd array
20	Division by zero
21	Illegal direct
22	Type mismatch
23	String too long
24	File data
25	Formula too complex
26	Can't continue error
27	Undef'd function
28	Verify
29	Load
30	Break

Riquadro 2. Codici di errore.



## Istruzione ON ERROR GOTO nn per il VIC 20 e C 64

Molti dei possessori di un Commodore 64 si saranno accorti della stretta somiglianza non solo estetica con il VIC 20.

Questa parentela investe l'interprete BASIC (v.2.0) assolutamente identico nei due computer anche se nel 64 è stato rilocato a partire da \$A000 (nel VIC parte da \$C000). La mappa delle routine di KERNAL è, invece, assolutamente uguale in ambedue. Molto importante è anche il fatto che la pagina zero sia rimasta invariata ad eccezione delle locazioni \$00, 01, 02 che non contengono più il vettore di salto dell'istruzione BASIC USR ora posto in \$310, 311, 312.

Grazie a questa compatibilità negli articoli futuri cercherò, dove possibile, di riferirvi ad entrambi i computer.

Ecco, intanto, le correzioni degli articoli già pubblicati per adattarli al 64.

Num. 5 DISABILITAZIONE TASTO R/S: valido anche per il 64.

Num. 8-9 GESTIONE CURSORE: cambiare nel testo e nel programma SYS58759 con SYS65520 e nel programma le due linee seguenti.

```
60260 CX = CX - 1 : IFCX = -1 THEN CX = 39 : CY = CY - 1
60280 CX = CX + 1 : IFCX = -40 THEN CX = 0 : CY = CY + 1)
```

### Riquadro 3. Compatibilità tra VIC 20 e C 64.

```
1 REM CARICA LA ROUTINE DI ON ERROR GOTO LNX
10 POKE56,PEEK(56)-1:POKE52,PEEK(52)-1
20 IN=PEEK(56)*256+PEEK(55)
30 FORI=0TO99:READN:POKEIN+I,N:NEXT
32 REM
35 REM ATTIVA LA ROUTINE
37 REM
40 POKE768,PEEK(55):POKE769,PEEK(56)
42 REM
45 REM NUM.LINEA A CUI SALTARE
47 REM
50 LNX=1000
60 LOAD"MENU",8
1000 REM
1005 REM LEX=LINEA DELL'ERRORE
1010 REM ST=CODICE ERRORE
1015 REM
1020 IFLEX<60THEN1100
1030 IFST<STHEN1060
1040 PRINT"ACCENDI IL LETTORE"
1045 PRINT"FLOPPY E PREMI 'CRET'"
1050 GETA$:IFA#<CHR$(13)THEN1050
1055 GOT060
1060 IFST<4THEN1100
1070 PRINT"NON TROVO IL PROGRAMMA"
1080 PRINT"CAMBIA DISCHETTO E"
1085 PRINT"PREMI 'CRET'"
1090 GETA$:IFA#<CHR$(13)THEN1090
1095 GOT060
1100 PRINT"ERRORE NUM"ST
1110 PRINT"NELLA LINEA"LEX
```

READY.

Listato 1. Esempio di uso della routine descritta nell'articolo. Vanno aggiunte le linee del listato 2 o 3.

```
1000 DATA138,133,144,164,058,200,240,080,169,128
1010 DATA872,169,204,133,069,169,197,133,070,160
1020 DATA000,032,235,208,160,000,165,058,145,071
1030 DATA208,165,057,145,071,169,204,133,069,169
1040 DATA206,133,070,160,000,032,235,208,194,160
1050 DATA000,177,071,133,021,200,177,071,133,020
1060 DATA165,043,166,044,032,023,198,144,017,165
1070 DATA895,233,001,133,122,165,096,233,000,133
1080 DATA123,104,104,076,234,199,162,017,076,058
1090 DATA196,234,234,234,234,098,067,068,230
```

### Listato 2. Data per il VIC 20.

```
1000 DATA138,016,003,076,116,164,133,144,164,058
1010 DATA200,240,080,169,128,072,169,204,133,069
1020 DATA169,197,133,070,160,000,032,235,176,160
1030 DATA000,165,058,145,071,200,165,057,145,071
1040 DATA169,204,133,069,169,206,133,070,160,000
1050 DATA032,235,176,164,160,000,177,071,133,021
1060 DATA200,177,071,133,020,165,043,166,044,032
1070 DATA023,166,144,017,165,095,233,001,133,122
1080 DATA165,095,233,000,133,123,104,104,076,234
1090 DATA167,162,017,076,058,164,234,234,234,234
```

### Listato 3. Data per il Commodore 64.

```
0400 ;ROUTINE DI INTERCETTAZIONE
0400 ;ERRORI
0400
0400 BA TXA ;
0401 B5 90 STA 590 ; codice errore in ST
0403 A4 3A LDY 53A ;
0405 C8 INY ; controlla se e' un errore in
0406 F0 50 BEQ 5045H ; modo diretto
0408 A9 80 LDA 800 ;
040A 48 PHA ;
040B A9 CC LDA 8CC ;
040D 85 45 STA 545 ;
040F A9 CE LDA 8CC ; cerca la variabile LEX e ne
0411 09 45 LDA 8CC ;
0413 80 00 LDY 800 ; mette l'indirizzo in 547,48
0415 20 EB 00 JSR 5D0EH ;
041B A0 00 LDY 800 ;
041A A5 3A LDA 53A ;
041C 91 47 STA 547,Y ;
041E C8 INY ;
041F A5 39 LDA 539 ; copia in LEX il numero della
0421 91 47 STA 547,Y ; ultima linea BASIC eseguita
0423 A9 CC LDA 8CC ;
0425 85 45 STA 545 ;
0427 A9 CE LDA 8CE ; cerca la variabile LNX
0429 85 46 STA 546 ;
042B A0 00 LDY 800 ;
042D 20 EB 00 JSR 5D0EH ;
0430 68 PLA ;
0431 A0 00 LDY 800 ;
0433 B1 47 LDA 547,Y ; copia il numero di linea da
0435 05 15 STA 515 ; cui riprendere l'esecuzione
0437 C8 INY ; del programma in 514,15
0438 B1 47 LDA 547,Y ;
043A 89 14 STA 514 ;
043C A5 2B LDA 52B ; cerca la linea con il numero
043E A6 2C LDA 52C ; contenuto in 514,15 e ne
0440 20 17 CB JSR 5C617 ; tiene l'indirizzo in 55F,60.
0442 80 11 BEQ 2056E ; c'è se non la trova
0445 A5 5F LDA 55F ;
0447 E9 01 SBC 01 ;
0449 85 7A STA 57A ; aggiorna il puntatore BASIC
044B A5 60 LDA 560 ; all'indirizzo di inizio del-
044D E9 00 SBC 00 ; la linea cercata prima
044F 85 7B STA 57B ;
0451 68 PLA ; riprende l'esecuzione del
0452 68 PLA ; del programma BASIC
0453 4C EA C7 JMP 5C7EA ;
0456 A2 11 LDX 811 ; cod. errore: UNDEF'D STATEMENT
0458 4C 3A C4 JMP 5C43A ; salta alla ERROR ROUTINE
```

Listato 4. Disassemblato della routine in linguaggio macchina.



TEXAS TI99/4A

## Rette ad alta risoluzione

Sergio Borsani

Consideriamo brevemente l'output su video del TI99/4A.

Il computer utilizza 24 righe e 32 colonne; in ognuna di queste posizioni può essere posto un carattere. Ad esempio l'istruzione CALL HCHAR (12, 16, 42), può essere usata come comando immediato che nei programmi BASIC fa apparire un asterisco (codice ASCII 42) nella posizione individuata dalla riga 12 e dalla colonna 16. Ricordate che le righe sono numerate dall'alto verso il basso e le colonne da sinistra a destra. La superficie occupata da un carattere è composta da una matrice quadrata formata da  $8 \times 8 = 64$  punti video (pixel). È possibile definire nuovi caratteri con l'istruzione CALL CHAR, specificando una stringa di 16 caratteri in codice esadecimale (vedi manuale), ognuno dei quali è associato ad un gruppo di 4 pixel. Ad esempio, CALL CHAR (128, "FF83858991A1C1FF") definisce un nuovo carattere che ha la forma di un quadratino attraversato da una diagonale; per farlo apparire bisogna usare CALL HCHAR; ricordate inoltre che i codici esadecimali posti tra virgolette in CALL CHAR vanno scritti con le lettere maiuscole.

Come molti sanno, l'equazione esplicita della retta è:  $y = m \times x + n$ , dove  $m$  rappresenta il coefficiente angolare ed  $n$  l'ordinata all'origine. Semplificando,  $m$  determina l'inclinazione della retta,  $n$  la fa spostare verso l'alto o verso il basso. Se  $n$  è uguale a 0 la retta passa per l'origine degli assi cartesiani.

Il modo più semplice per disegnare una retta con il computer è quello di stampare una serie di asterischi, uno per ogni colonna, usando l'istruzione CALL HCHAR. Molte versioni del BASIC sono sprovviste di tale comando, nel qual caso si ricorre al PRINT TAB.

Comunque, una retta rappresentata da una serie di asterischi sembra tale solo per certe inclinazioni: per  $m = 0$  (retta orizzontale) ed  $m = 1$  (retta a 45°); in tutti gli altri casi assume un poco gradito aspetto a gradini.

Questo può essere accettabile su monitor ad 80 colonne ma non lo è assolutamente su uno a 32.

Se non avete mai provato, fate girare il programma del listato 1.

Per ottenere una retta ad alta risoluzione bisogna essere in grado di gestire i singoli punti video. In alcuni computer è sufficiente il comando LINE ( $X_1,$

$Y_1$ ) - ( $X_2, Y_2$ ) per ottenere un segmento di retta che congiunge due punti video.

Con il TI99/4A la cosa è possibile solo creando una routine di assembler e sappiamo che per disporre di tale linguaggio è necessario avere il sistema a dischi.

Apparentemente sembra impossibile ottenere lo stesso risultato in BASIC, il programma presentato in questo numero dimostra tuttavia il contrario.

Il prezzo pagato, però, è molto alto e si traduce in una lenta esecuzione, al punto che il programma non può essere usato per normali applicazioni ma va piuttosto considerato come un "pezzo da museo".

All'inizio dell'esecuzione viene chiesto di introdurre i valori di  $m$  e di  $n$ . Il coefficiente angolare può assumere valori positivi o negativi, è opportuno tuttavia che non oltrepassi un certo limite; già per  $m = 3$  i punti che formano la retta non sono più adiacenti tra di loro; perfezionando il programma sarebbe tuttavia possibile eliminare questo inconveniente. Anche  $n$  può assumere valori positivi o negativi, ma se questi sono troppo alti la retta cade fuori campo e non viene visualizzata. Alla riga 330 viene definita l'equazione della retta:  $y$  è l'ordinata, in numero di pixel,  $x$  è l'ascissa;  $n$  è stato moltiplicato per 10 in modo da ottenere una immagine "ingrandita" del grafico. Nelle linee 570-690 viene individuato il carattere più vicino all'asse verticale che viene attraversato dalla retta;  $NX$  ed  $NY$  sono il numero di caratteri interposti tra questo e gli assi.

Le linee 700-1070 contengono la routine che determina quali pixel, all'interno del carattere, devono essere messi in posizione ON; il compito è svolto da un ciclo interno FOR ... NEXT che esplora i pixel a gruppi di 4 e genera un codice esadecimale, e da un ciclo esterno FOR ... NEXT di 16 passi che compone la stringa atta a definire l'intero carattere.

La variabile MAX memorizza la distanza in pixel dell'ultimo punto dell'asse verticale man mano che procede la costruzione della retta, mentre T numera i caratteri che di volta in volta vengono visualizzati.

In sostanza il programma, non potendo gestire i punti video indipendentemente uno dall'altro, si incarica di esplorarli, raggrupparli, generare da questi una stringa in codice esadecimale, definire un carattere stamparlo quindi nella posizione opportuna.

Ogni ciclo completo viene effettuato in circa 11", tuttavia una retta può essere formata da più di 20 caratteri, il che comporta un tempo totale di esecuzione intorno ai 3 minuti, in verità troppo alto se si dovesse usare la routine in un programma applicativo.

Tengo a precisare che tale lentezza non è imputabile al TI99/4A, che in alcuni benchmarks da me effet-

## Rette ad alta risoluzione.

tuati si è generalmente dimostrato all'altezza di molti veri personal per quanto riguarda la velocità a livello di CPU, ma esclusivamente all'uso del BASIC come linguaggio di programmazione in questa particolare applicazione.

Sapreste, cambiando solo poche righe, ottenere grafici di altre funzioni? ■

Listato 1. Esempio di retta rappresentata mediante una serie di asterischi.

```

100 CALL CLEAR
110 PRINT TAB(4); "EQUAZIONE ";
120 PRINT "DELLA RETTA"
130 PRINT TAB(10); "Y=M*X+N"
140 INPUT "M = "; M
150 INPUT "N = "; N
160 DEF Y=M*X+N
170 CALL CLEAR
180 FOR X=1 TO 32
190 IF Y<1 THEN 220
200 IF Y>24 THEN 220
210 CALL HCHAR(25-Y, X, 42)
220 NEXT X
230 END
    
```

Listato 2. Programma retta ad alta risoluzione.

```

100 REM *****
110 REM * RETTE *
120 REM * AD ALTA *
130 REM * RISOLUZIONE *
140 REM *****
150 CALL CLEAR
160 DIM A$(50)
170 T=1
180 MAX=0
190 PRINT " RETTE"
200 CALL CHAR(96, "01010101010101")
210 CALL CHAR(97, "FF")
220 CALL CHAR(98, "01010101010101")
230 CALL CHAR(99, "01010101010101")
240 CALL CHAR(100, "FF404")
250 CALL CHAR(101, "FF101")
260 CALL CHAR(104, "0")
270 FOR C=10 TO 16
280 CALL COLOR(C, 2, 16)
290 NEXT C
300 PRINT " Y=M*X+N"
310 INPUT "COEFF. ANGOLARE? (M) ":M
320 INPUT "ORD. ALL'ORIGINE? (N) ":N
330 DEF Y=INT(M*X+N*10)
340 CALL CLEAR
350 FOR RIGA=3 TO 16
360 CALL HCHAR(RIGA, 10, 104, 14)
370 NEXT RIGA
380 CALL HCHAR(17, 10, 97, 14)
390 CALL HCHAR(3, 9, 48, 14)
400 CALL HCHAR(18, 8, 48)
410 CALL HCHAR(17, 16, 100)
420 CALL HCHAR(18, 16, 53)
430 CALL HCHAR(17, 22, 101)
440 CALL HCHAR(18, 22, 49)
450 CALL HCHAR(18, 23, 48)
460 CALL HCHAR(10, 9, 98)
470 CALL HCHAR(10, 8, 53)
    
```

Segueo listato 2.

```

480 CALL HCHAR(4, 9, 99)
490 CALL HCHAR(4, 7, 49)
500 CALL HCHAR(4, 5, 48)
510 EQ$="Y=("&STR$(M)&")*X+("&STR$(N)&")"
520 IF LEN(EQ$)>20 THEN 570
530 FOR I=1 TO LEN(EQ$)
540 Z=ASC(SEG$(EQ$, I, 1))
550 CALL HCHAR(20, 9+I, Z)
560 NEXT I
570 X=1
580 IF X>112 THEN 1080
590 IF Y>112 THEN 620
600 IF Y<1 THEN 620
610 GOTO 640
620 X=X+1
630 GOTO 580
640 NY=INT(Y/8)
650 IF NY<Y/8 THEN 670
660 NY=NY-1
670 NX=INT(X/8)
680 IF NX<X/8 THEN 710
690 NX=NX-1
700 REM *DEFINIZIONE E STAMPA CARATTERE*
710 J=8
720 NXX=8*NX
730 NYY=8*NY
740 FOR F=1 TO 16
750 S=0
760 Y1=NYY+J
770 IF INT(F/2)=F/2 THEN 800
780 W=0
790 GOTO 810
800 W=4
810 FOR K=1 TO 4
820 X=NXX+K+W
830 IF Y1=INT(Y) THEN 860
840 P=0
850 GOTO 890
860 F=2^(4-K)
870 IF X<MAX THEN 890
880 MAX=X
890 S=S+P
900 NEXT K
910 IF W=0 THEN 930
920 J=J-1
930 IF S>9 THEN 960
940 COD$=STR$(S)
950 GOTO 970
960 COD$=CHR$(55+S)
970 IF F<>1 THEN 1000
980 CAR$=COD$
990 GOTO 1010
1000 CAR$=CAR$&COD$
1010 NEXT F
1020 A$(T)=CAR$
1030 CALL CHAR(104+T, A$(T))
1040 CALL HCHAR(16-NY, 10+NX, 104+T)
1050 T=T+1
1060 X=MAX+1
1070 GOTO 580
1080 V$="VUOI CONTINUARE? (S/N) "
1090 FOR I=1 TO LEN(V$)
1100 L=ASC(SEG$(V$, I, 1))
1110 CALL HCHAR(24, 3+I, L)
1120 NEXT I
1130 CALL KEY(0, KEY, STATUS)
1140 IF STATUS=0 THEN 1130
1150 CALL CLEAR
1160 IF KEY=83 THEN 150
1170 IF KEY=115 THEN 150
1180 END
    
```

TI99/4A HOME COMPUTER

## Videografici

Sergio Borsani

In questo stesso numero, mostro come sia possibile, con una routine in BASIC, ottenere il grafico di una retta ad alta risoluzione, anche se i tempi di esecuzione sono molto lenti. Il maggiore inconveniente è che i pixel video, pur essendo indirizzabili individualmente, sono condizionati dagli altri punti-video che formano un carattere. Pertanto la prassi consiste nel definire un carattere indirizzando al suo interno i punti mediante una stringa in codice esadecimale (CALL CHAR(N,A\$)) e poi indirizzare sul video tutto il carattere mediante una CALL HCHAR(X,Y,N), dove X è il numero di riga, Y il numero di colonna e N il numero del carattere precedentemente definito.

Per la grafica ad alta risoluzione è consigliabile munirsi di carta quadrettata, con quadrati suddivisi in 8 x 8 quadratini (purtroppo la carta millimetrata non è del tutto adatta). Come esempio immaginiamo di voler definire un carattere che riproduce il cavallo del gioco degli scacchi (vedi figura 1).

Il disegno va diviso in due parti con una linea verticale ed in tutto si ottengono 16 blocchi di quattro pixel ciascuno. I blocchi vanno letti, normalmente, da sinistra a destra e dall'alto in basso. Essi si traducono poi in codice esadecimale, il manuale (pag. 84) è abbastanza chiaro a tale proposito. Il disegno del nostro esempio viene definito con l'istruzione CALL CHAR(128,"083C3E0E1E1E0C3E").

La stringa che compare nell'istruzione è composta da 16 caratteri esadecimale, tanti sono i blocchi, tuttavia eventuali zeri finali possono essere trascurati; ad esempio, volendo in posizione ON tutti i pixel di un carattere sarà sufficiente l'istruzione CALL CHAR(129,"F") e poi dare lo stesso colore al carattere ed allo sfondo con CALL COLOR(13,9,9). Il numero 9 si riferisce al rosso.

Le cose si complicano quando si vogliono creare immagini ad alta definizione composte da molti caratteri. In un mio programma ho riprodotto la cartina d'Italia, comprese le isole, in dimensioni tali da occupare quasi tutto lo schermo.

Per far questo ho dovuto definire circa 80 caratteri e posizionarli opportunamente sullo schermo. Quando si trattano caratteri ridefiniti in numero così elevato bisogna essere consapevoli che non possono essere utilizzati per altri scopi. Paradossalmente, se si ridefinisce il carattere con codice ASCII 32 (space), quando poi nel programma si dà l'istruzione CALL CLE-

AR, tutto lo schermo si riempie con il nuovo carattere invece di pulirsi.

Una volta definiti, i caratteri vanno indirizzati sul video. Se sono in gran numero non sarà necessario ripetere decine di volte l'istruzione CALL HCHAR, ma sarà da preferirsi l'uso di un ciclo FOR NEXT come nel seguente esempio:

```
...
500 DATA 12, 15, 128, 12, 16, 129 ...
510 RESTORE 500
520 FOR J = 1 TO ...
530 READ X, Y, N
540 CALL HCHAR (X,Y, N)
550 NEXT J
...
```

I dati devono essere nell'ordine: numero di riga, di colonna, di carattere.

In ogni caso la creazione di immagini composte da molti caratteri comporta sempre una notevole perdita di tempo. Sarebbe certamente vantaggioso disegnare direttamente sul video, tramite la tastiera, lasciando al computer il compito di determinare la stringa di caratteri esadecimale.

Questo è quanto ho realizzato nel programma che presento nel listato 1.

Selezionato da menu il tasto 1, compaiono sul video un quadrato bianco ed un riquadro. Il primo rappresenta un carattere ingrandito, il secondo lo spazio in cui andrà collocato il disegno ad alta risoluzione che si vuole realizzare. Il quadrato bianco viene spazzato da un cursore. Premendo i tasti 1 e 0, il cursore lascia dietro di sé rispettivamente un punto nero (pixel ON) o un punto bianco (pixel OFF). Se si commette un errore si può annullare l'ultima parte del lavoro con SHIFT(-). Completata l'operazione appare la scritta PROCEDO? Ad una risposta affermativa compare il numero del carattere e la stringa in codice esadecimale che lo definisce, inoltre appare il carattere in dimensioni reali nel riquadro. (Figura 2).

Azionando i tasti muniti di freccia (E = alto, S = sinistra, D = destra, X = basso) si può quindi collocare quest'ultimo nella posizione più opportuna. Premendo (1) si procede alla definizione di un secondo carattere.

Continuando in questo modo si aggiungono nuovi caratteri e si accostano a quelli precedentemente definiti fino a formare una figura complessa ad alta risoluzione. Al termine bisogna premere SHIFT(+).

Se si limitasse a questo il programma avrebbe solo un valore didattico, invece esso permette di memorizzare su nastro (o, modificando le OPEN, su dischetto) tutta l'immagine creata per poterla poi richiamare e riutilizzare in un altro programma; anzi, in questo secondo programma, potrebbero trovar posto più

Videografici

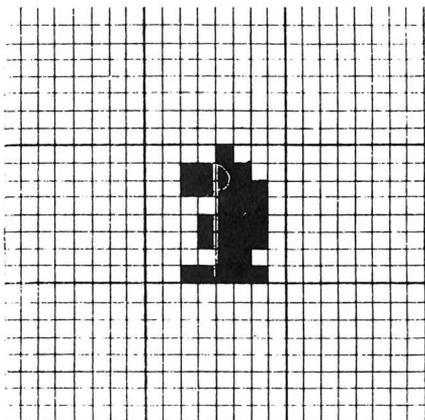


Figura 1. Rappresentazione del cavallo nel gioco degli scacchi. Viene definito con l'istruzione CALL CHAR(128, "083C3E0E1E1E0C3E").

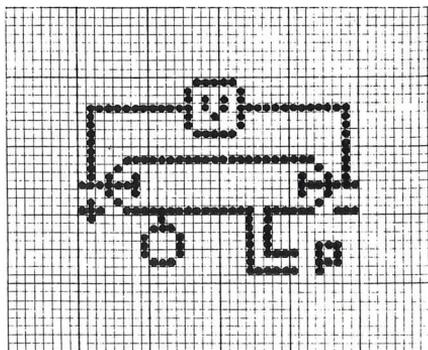


Figura 3. Disegno in alta risoluzione formato da 12 caratteri. Con Videograf si creano immagini come questa, si memorizzano e si riutilizzano in altri programmi.

disegni, richiamati successivamente con le stesse modalità. (Figura 3).

Nella prima fase la "finestra" che contiene il disegno viene esplorata con una istruzione CALL GCHAR che restituisce i numeri dei caratteri presenti; i dati,

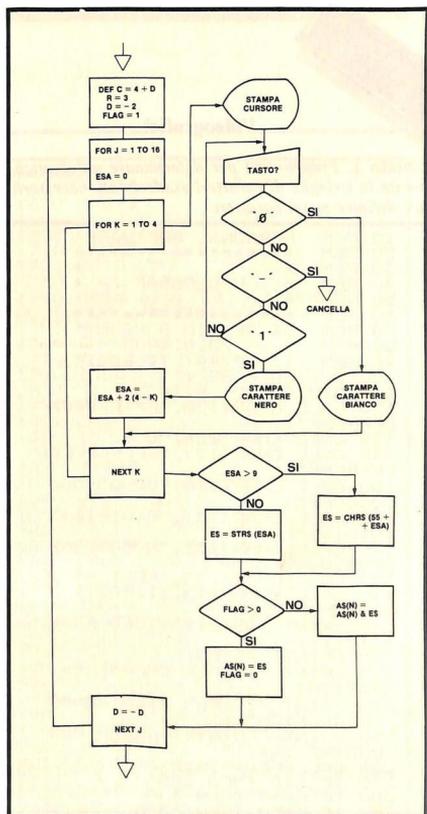


Figura 2. Routine per generare la stringa di caratteri esadecimale che definiscono un carattere. Il ciclo K esplora blocchi di 4 pixel; se  $K = 1$ ,  $ESA = ESA + 2^4 (4 - K)$ , cioè 8; per  $K = 2$ ,  $ESA = 4$ , ecc.

comprendenti le coordinate, vengono posti nella matrice B(30,3) e successivamente registrati su nastro. Per procedere alla registrazione, come ho già detto, bisogna premere SHIFT(+) appena terminato il disegno. L'uso di questa matrice si rende necessario poiché, quando si dà la OPEN in output, compaiono le istruzioni per l'uso del registratore e il disegno si sposta verso l'alto fuori campo cosicché non può più essere riprodotto.

Per richiamare l'immagine è necessario inserire nel nuovo programma la parte finale di questo listado che inizia con OPEN ≠ 10:"CSI",SEQUENTIAL, INPUT, INTERNAL, FIXED (linea 1740) e specificare le coordinate relative per porre il disegno nella posizione opportuna, specificando eventualmente il colore.

**Videografici**

Listato 1. Programma per determinare automaticamente le stringhe di caratteri esadecimale, necessarie per definire nuovi caratteri.

```

10 REM PERSONAL SOFTWARE
20 REM *****
30 REM *
40 REM * VIDEOGRAF *
50 REM *
60 REM *****
70 REM DI SERGIO BORSANI
80 REM 28.6.1983
90 REM TI-99/4A TI BASIC
100 OPTION BASE 1
110 DIM A$(30), B(24,3)
120 CALL CHAR(128,"FFFFFFFF
FFFFFFFF")
130 CALL CHAR(126,"0")
140 CALL CHAR(127,"FFFFFFFF
FFFFFFFF")
150 CALL CHAR(130,"00000000
000000FF")
160 CALL CHAR(131,"01010101
01010101")
170 CALL CHAR(132,"80808080
80808080")
180 CALL CHAR(133,"FF")
190 CALL COLOR(12,11,16)
200 CALL CLEAR
210 PRINT TAB(8);"VIDE GRAF
ICI":::
220 PRINT "PREMI:"
230 PRINT "1. PER CREARE UN
GRAFICO"
240 PRINT "2. PER RICHIAMARE
UN GRAFICO"
250 PRINT "FCTN(4)=CLEAR PER
USCIRE":::
260 CALL KEY(0,K,S)
270 IF S=0 THEN 260
280 IF K=49 THEN 300
290 IF K=50 THEN 1730
300 CALL CLEAR
310 CALL HCHAR(3,16,130,12)
320 CALL VCHAR(4,15,131,12)
330 CALL VCHAR(4,28,132,12)
340 CALL HCHAR(16,16,133,12)
350 N=0
360 N=N+1
370 IF N<24 THEN 450
380 FINE$="FINE CARATTERI"
390 FOR J=1 TO LEN(FINE$)
400 L=ASC(SEG$(FINE$,J,1))
410 CALL HCHAR(18,15+J,L)
420 NEXT J
430 N=23
440 GOTO 1280
450 CALL HCHAR(20,1,32,16)
460 FOR R=4 TO 11
470 CALL HCHAR(R,3,126,8)
480 NEXT R
490 R=3
500 D=-2
510 DEF C=4+D
520 FLAG=1
530 FOR J=1 TO 16
540 ESA=0
550 FOR K=1 TO 4
560 CALL HCHAR(R+INT((J+1)/2
),C+K,127)

```

Segue figura 1.

```

570 CALL KEY(0,KEY,S)
580 IF S=0 THEN 570
590 IF KEY=48 THEN 650
600 IF KEY=45 THEN 450
610 IF KEY<>49 THEN 570
620 CALL HCHAR(R+INT((J+1)/2
),C+K,128)
630 ESA=ESA+2^(4-K)
640 GOTO 660
650 CALL HCHAR(R+INT((J+1)/2
),C+K,126)
660 NEXT K
670 IF ESA>9 THEN 700
680 E$=STR$(ESA)
690 GOTO 710
700 E$=CHR$(55+ESA)
710 IF FLAG=0 THEN 750
720 A$(N)=E$
730 FLAG=0
740 GOTO 760
750 A$(N)=A$(N)&E$
760 D=D
770 NEXT J
780 DOM$="PROCED0?"
790 FOR J=1 TO LEN(DOM$)
800 L=ASC(SEG$(DOM$,J,1))
810 CALL HCHAR(20,2+J,L)
820 NEXT J
830 CALL KEY(0,KEY,S)
840 IF S=0 THEN 830
850 IF KEY=83 THEN 890
860 IF KEY=78 THEN 450
870 GOTO 830
880 CALL HCHAR(20,1,32,32)
890 NC=136+N
900 RIGA$="CHR$ "&STR$(NC)&"
&CHR$(34)&A$(N)&CHR$(34)
910 CALL HCHAR(1,1,32,32)
920 FOR J=1 TO LEN(RIGA$)
930 L=ASC(SEG$(RIGA$,J,1))
940 CALL HCHAR(1,2+J,L)
950 NEXT J
960 X=4
970 Y=22
980 CALL CHAR(NC,A$(N))
990 CALL HCHAR(X,Y,NC)
1000 CALL KEY(0,KEY,S)
1010 IF S=0 THEN 1000
1020 XVECCHIO=X
1030 YVECCHIO=Y
1040 IF KEY=69 THEN 1110
1050 IF KEY=88 THEN 1140
1060 IF KEY=93 THEN 1170
1070 IF KEY=68 THEN 1200
1080 IF KEY=49 THEN 1230
1090 IF KEY=43 THEN 1270
1100 GOTO 1000
1110 IF X=4 THEN 1000
1120 X=X-1
1130 GOTO 1040
1140 IF Y=1240 THEN 1000
1150 X=X+1
1160 GOTO 1040
1170 IF Y=16 THEN 1000
1180 Y=Y-1
1190 GOTO 1040
1200 IF Y=27 THEN 1000
1210 Y=Y+1
1220 GOTO 1040
1230 GOTO 360
1240 CALL HCHAR(XVECCHIO,YVE
CCHIO,32)
1250 CALL HCHAR(X,Y,NC)
1260 GOTO 1000
1270 REM *** REGISTRA ***
1280 REG$="VUOI REGISTRARE?
(S/N) "
1290 CALL HCHAR(20,1,32,32)

```

**PERSONAL SOFTWARE**

## Videografici

Seguito figura 1.

```

1300 FOR J=1 TO LEN(REG#)
1310 L=ASC(SEG$(REG$,J,1))
1320 CALL HCHAR(20,5+J,L)
1330 NEXT J
1340 CALL KEY(0,KEY,S)
1350 IF S=0 THEN 1340
1360 IF KEY=78 THEN 200
1370 IF KEY=83 THEN 1390
1380 GOTO 1340
1390 RR$="RUNNING"
1400 FOR J=1 TO LEN(RR#)
1410 L=ASC(SEG$(RR$,J,1))
1420 CALL HCHAR(24,11+J,L)
1430 NEXT J
1440 CTR=0
1450 FOR J=1 TO 12
1460 FOR K=1 TO 12
1470 CALL GCHAR(3+J,15+K,CX)
1480 IF CX=32 THEN 1530
1490 CTR=CTR+1
1500 B(CTR,1)=J
1510 B(CTR,2)=K
1520 B(CTR,3)=CX
1530 NEXT K
1540 NEXT J
1550 CALL CLEAR
1560 IF CTR>0 THEN 1610
1570 PRINT "NON C'E' NIENTE!"
1580
1580 FOR T=1 TO 500
1590 NEXT T
1600 GOTO 110
1610 OPEN #10:"CS1",SEQUENTI
AL,OUTPUT,INTERNAL,FIXED
1620 PRINT #10:N
1630 FOR J=1 TO N
1640 PRINT #10:B(J,1);B(J,2)
;B(J,3)
1650 NEXT J
1660 J=1
1670 PRINT #10:A$(J);A$(J+1)

```

Seguito figura 1.

```

;A$(J+2)
1680 J=J+3
1690 IF J>N THEN 1710
1700 GOTO 1670
1710 CLOSE #10
1720 GOTO 200
1730 CALL CLEAR
1740 OPEN #10:"CS1",SEQUENTI
AL,INPUT,INTERNAL,FIXED
1750 N=0
1760 INPUT #10:N
1770 FOR J=1 TO N
1780 INPUT #10:B(J,1),B(J,2)
;B(J,3)
1790 NEXT J
1800 J=1
1810 INPUT #10:A$(J),A$(J+1)
;A$(J+2)
1820 J=J+3
1830 IF J>N THEN 1850
1840 GOTO 1810
1850 CLOSE #10
1860 CALL CLEAR
1870 FOR J=1 TO N
1880 CALL CHAR(136+J,A$(J))
1890 NEXT J
1900 PRINT "COORDINATE":
1910 INPUT "RIGA?":X0
1920 INPUT "COLONNA?":Y0
1930 CALL CLEAR
1940 FOR J=1 TO N
1950 RIG=X0+B(J,1)
1960 COL=Y0+B(J,2)
1970 IF RIG<1 THEN 2020
1980 IF RIG>24 THEN 2020
1990 IF COL<1 THEN 2020
2000 IF COL>32 THEN 2020
2010 CALL HCHAR(RIG,COL,B(J,3))
2020 NEXT J
2030 GOTO 2030
2040 END

```

# TELEMATICA

Dal viewdata all'office automation

Tutti oggi parlano di telematica, di società dell'informazione, di banche dati.

Ma cosa è la telematica? Un insieme di servizi di videoterminali e trasmissione di dati e testi. Innanzitutto la videoinformazione. Essa rappresenta un servizio che, utilizzando le reti telematiche pubbliche, permette ad un qualsiasi utente, dotato di un televisore a colori adatto, di richiedere e ricevere informazioni memorizzate su opacità (banche di dati (Videotext e Televideo)). Poi vi sono i servizi pubblici per la trasmissione di testi scritti da terminale a terminale ed i fac-simile. Essi sono basati, fra l'altro, per la realizzazione della "posta elettronica".

Le applicazioni della telematica sono infinite ed in parte ancora da scoprire. Esse è, innanzitutto, un nuovo e potente "medium" nel campo della comunicazione e dell'informazione, ma è

anche lo strumento principale che rivoluzionerà l'organizzazione e la produttività del lavoro di ufficio, per realizzare quello che si chiama "office automation".

Questo libro intende dare un impulso alla conoscenza della telematica, e si prefigge di offrire al lettore un panorama dei problemi connessi con questa disciplina e con i relativi aspetti applicativi. Le caratteristiche dell'esposizione fanno sì che il volume possa proporsi indifferentemente all'esperto ESP e di organizzazione, quanto allo studioso che si accosta per la prima volta a questa materia: l'esperto troverà un sicuro riferimento per la risoluzione di problemi teorici e pratici, mentre lo studioso troverà, in una forma organica, i principi fondamentali indispensabili per la conoscenza delle varie problematiche.

di **Riccardo Glucksmann**  
Cod. 518D Pag. 166  
L. 19.000

SCONTO 20% AGLI ABBONATI  
FINO AL 28-2-84

### Sommario

Telematica e suo sviluppo - Evoluzione delle telecomunicazioni per lo sviluppo della telematica - Reti per telecomunicazioni - Reti di calcolatori e banche dati - Videotex e Teletext - Altri nuovi servizi di telematica - Funzionalità del sistema videotex - Sviluppo del videotex nel mondo - Telematica in Italia - Sviluppo delle comunicazioni - Applicazioni della telematica - Comunicazioni di massa e aspetti socio-economici e giuridici.

Potete acquistare il suddetto libro nelle migliori librerie oppure scrivendo direttamente a: **Gruppo Editoriale Jackson - Divisione Libri - Via Rosellini, 12 20124 Milano**



ZX-SPECTRUM

**La funzione SCREEN\$: come utilizzarla anche per i caratteri grafici**

Marcello Spero

La particolare architettura dell'area video nello ZX Spectrum rende impossibile l'uso di PEEK e POKE per una gestione veloce dell'immagine nei programmi di movimento, come avviene con lo ZX81.

Questa situazione è dovuta alla soluzione adottata dalla Sinclair, che per migliorare la velocità di elaborazione dello Spectrum, ha tolto alla CPU il compito di visualizzare i dati ( che nello ZX81 occupa circa i quattro quinti del tempo totale). Questo compito è ora demandato ad un circuito (ULA) di tipo non intelligente, capace quindi solo di prelevare i dati e trasferirli allo schermo TV. La zona della memoria ove risiedono le informazioni per il video contiene perciò non semplicemente i codici dei caratteri ma la loro immagine completa formata da una "griglia" di punti già disposti nell'ordine in cui dovranno essere prelevati, ordine che rispecchia il percorso del pannello elettronico sullo schermo quando viene composta l'immagine.

Da un lato questa soluzione ha permesso l'adozione della grafica ad alta risoluzione, ma dall'altro costringe l'utilizzatore a tornare a tecniche indirette di manipolazione del video. Per quanto riguarda l'invio di caratteri la soluzione è l'uso di PRINT AT, la cui velocità di esecuzione è comunque superiore a quella di una POKE sullo ZX81. Per la lettura dello schermo, invece, è stata creata una funzione apposita, SCREEN\$, che usata nella forma SCREEN\$(m,n) riporta il carattere presente in riga m, colonna n dello schermo.

Questa funzione, peraltro utilissima, ha però un difetto che si fa sentire in quei programmi che usano i caratteri grafici definibili; questi caratteri, infatti, non vengono riconosciuti dalla funzione, che riporta al loro posto una stringa vuota.

Il motivo di questa carenza risiede nel procedimento usato da SCREEN\$ per il riconoscimento di un carattere; ciò avviene in due stadi: recupero ed ordinamento degli otto byte che compongono ogni carattere nell'area video, e successiva ricerca all'interno del set di caratteri di una sequenza uguale. Osservando che il set di caratteri dello Spectrum inizia con lo spazio, CHR\$ 32, e termina con il simbolo °, CHR\$ 127, comprendendo le lettere maiuscole e minuscole,

i numeri e i caratteri speciali, ma non quelli grafici, comprendiamo l'impossibilità di riconoscere questi ultimi.

Per ovviare a questa carenza ci sono tre metodi: ricorrere a tecniche di verifica basate sul confronto fra coppie di coordinate (senz'altro piuttosto lente), usare la funzione ATTR (se i caratteri da leggere hanno tutti attributi diversi), o cercare di includere i caratteri grafici nel set. È appunto questa la tecnica che vi propongo.

Ogni volta che, nel corso delle elaborazioni, sorge la necessità di ricorrere al set di caratteri, questo viene trovato per mezzo della variabile di sistema CHAR\$(23606/7) che ne indica l'inizio o meglio, per essere più precisi, punta 256 byte prima del suo inizio.

A questa regola non sfugge SCREEN\$, che utilizza CHAR\$ come riferimento nella sua ricerca.

Il set utilizzato normalmente è quello in ROM, creato dallo Sinclair secondo lo standard ASCII, ma il manuale stesso accenna alla possibilità di crearne un altro, cui verrà fatto puntare CHAR\$.

Nel nostro caso questo set alternativo esiste già: è l'area della grafica definibile, la cui posizione ci viene indicata dalla variabile di sistema UDG (23675/6); quello che ci resta da fare è cambiare il contenuto di CHAR\$ perché punti a questo nuovo indirizzio.

Per questo bastano le seguenti istruzioni:

```
POKE 23606, PEEK 23675
POKE 23607, PEEK 23676-1
```

ricordandoci che CHAR\$ deve puntare 256 byte prima dell'inizio del set. A questo punto il primo carattere grafico sarà interpretato come CHR\$ 32 cioè il carattere corrispondente al tasto dello spazio, il secondo come CHR\$ 33 cioè al posto del !, e così via. Ovviamente nel listato del programma andranno indicati questi caratteri; quindi:

```
IF SCREEN$(15,8) = CHR$ 32 THEN ...
```

oppure:

```
IF SCREEN$(15,8) = " " THEN ...
```

e non:

```
IF SCREEN$(15,8) = CHR$ 144 THEN ...
```

per verificare la presenza in riga 15 e colonna 8 del carattere corrispondente alla A grafica.

Per semplificare la correlazione fra i caratteri del nuovo set e quelli indicati sui tasti può essere utile spostare CHAR\$ indietro di 65 caratteri, cioè di 65 \* 8 = 520 byte; in questo modo il primo carattere



## La funzione SCREEN\$: come utilizzarla anche per i caratteri grafici

grafico verrà a corrispondere alla a minuscola, il secondo alla b e così via. Si scriverà quindi:

IF SCREEN\$ (15,8) = "a" THEN ...

per indicare il carattere corrispondente alla A grafica, evitando possibili errori. In questo caso bisogna operare così:

POKE 23606, PEEK 23675-8  
POKE 23607, PEEK 23676-3

essendo  $8 + (3 \star 256) = 776 (520 + 256)$ .

Tenete presente che dopo aver sostituito il set, e fino a quando non lo avrete ripristinato con:

POKE 23606,0  
POKE 23607,0

vi sarà impossibile ottenere lettere, numeri o scritte di qualunque tipo e sarete esposti al rischio che un errore, interrompendo il programma prima che questo abbia avuto il tempo di riportare le cose alla normalità, vi lasci in un mondo "da incubo" in cui non potete leggere nè il messaggio di errore nè i caratteri che eventualmente scriverete. Se vi dovesse capitare questa avventura, state calmi e, soprattutto, non spegnete la macchina: non è necessario! Prendete invece un profondo respiro e, senza guardare il video, digitate lentamente e accuratamente (se dovete sbagliare, premete EDIT e ENTER per cancellare tutto e ricominciate da capo) le due POKE viste sopra. A me è capitato, e ne sono sempre uscito bene ...

Comunque, per minimizzare questo pericolo, è bene cambiare set solo quando è necessario e tornare al set normale subito dopo, con una serie di istruzioni del tipo:

POKE 23606, PEEK 23675-8  
POKE 23607, PEEK 23676-3  
IF SCREEN\$ (15,8) = "F" THEN ... (o qualsiasi altra istruzione che usi SCREEN\$)  
POKE 23606,0  
POKE 23607,0

facendo attenzione che dopo il THEN non ci sia un salto, perchè in questo caso verrebbero evitate le POKE seguenti.

Per concludere faccio notare che la funzione SCREEN\$ è utilizzabile anche nella forma SCREEN\$ (22,n) e SCREEN\$ (23,n), cioè per la parte bassa dello schermo, per cui se siete riusciti, in un modo o nell'altro, a scrivere qualcosa in queste due ultime righe potete utilizzarla tranquillamente. ■

# BA.SE s.n.c.

SOFTWARE HOUSE - Casella Postale 4  
13055 - Occhieppo Inferiore (VC)

Tel. 015/592730

## SONO DISPONIBILI PER **COMMODORE 64**

### ALTO MEDIOEVO

Una perfetta simulazione dell'economia medioevale. Rispetta le gerarchie feudali di vassallaggio e vi renderà esperti nell'arte di governare destreggiandovi tra guerre - carestie - epidemie - maltempo e inondazioni strutturato a economia di mercato permette elaborate politiche fiscali e speculazioni commerciali. Da 1 a 9 feudatari il migliore dei quali diventerà Re. Corredato di istruzioni.

**Lire 30.000** dischetto **Lire 25.000** cassetta

### ATOMO

Gestione simulata di impianto nucleare per la produzione di energia elettrica. Il pieno rispetto dei parametri reali rende il programma oltre che un gioco un modo per capire il funzionamento di un reattore nucleare. È la vostra condotta a determinare - rendimento - guasti ecc.

Necessarie buone doti di intuito e abilità - sarete comunque valutati dal calcolatore a fine impiego. Non aspettatevi giudizi molto lusinghieri (almeno all'inizio).

**Lire 30.000** dischetto **Lire 25.000** cassetta

### TORRE DI HANOI + OTHELLO

I - classici - finalmente anche per il Commodore 64.  
**Lire 30.000** dischetto **Lire 25.000** cassetta

**A disposizione per consulenze su**  
Software Applicativo - Automazione di Processi  
Soluzione dei Vs. problemi su Commodore 64  
Tel. 015/592730

**in vendita anche presso**  
TEOREMA - Via Losanna, 9 - Biella

Spedire in busta chiusa a:  
BA.SE s.n.c. - Casella Postale 4 - 13055 Occhieppo Inf. (VC)

Nome e Cognome \_\_\_\_\_

Indirizzo \_\_\_\_\_

Cap. \_\_\_\_\_ Città \_\_\_\_\_ Provincia \_\_\_\_\_

Ordine n° \_\_\_\_\_  Disco  Cassetta di Alto Medioevo

Ordine n° \_\_\_\_\_  Disco  Cassetta di Atomo

Ordine n° \_\_\_\_\_  Disco  Cassetta di Torre di Hanoi + Othello

Per un totale di Lire \_\_\_\_\_

Pagamento  Allegato assegno non trasf. sped. celerè  
 Contro assegno + Lire 2.000 s.s.



## Effetti sonori particolari con lo ZX Spectrum

*Il signor Mauro Carletti di Rimini (FO), ci propone queste interessanti note per un utilizzo musicale del versatile personal Sinclair.*

Con lo ZX Spectrum è possibile generare suoni con l'istruzione BASIC, BEEP durata - frequenza, dove la durata è espressa in secondi e la frequenza in semitoni con una escursione da -60 a +70 rispetto allo 0 il quale corrisponde al D0 centrale con frequenza di circa 216 Hz.

Questo comando però, come è facilmente intuibile dalla sua sintassi, permette di riprodurre solo una nota alla volta.

Non è possibile cioè, generare spari o altri effetti sonori che richiedano un'unica nota variabile in frequenza, oppure note in rapida successione più rapida delle possibilità dell'interprete BASIC.

Per avviare in parte a questo inconveniente si può ricorrere a delle routine in linguaggio macchina.

La routine che propongo è già codificata in codice decimale, ed è quindi pronta per essere *pokata* in memoria tramite il semplice programmino del listato 1.

Questa routine occupa 55 byte e genera, a seconda della locazione richiamata con RANDOMIZEUSR, due differenti suoni che ognuno può adattare alle proprie esigenze. Infatti durata, frequenza iniziale e numero di esecuzioni dipendono da valori alterabili con opportune POKE.

Prima di tutto bisognerà riservare, cambiando il puntatore di RAMTOP, lo spazio in memoria: per il 16 Kbyte si farà CLEAR 32543, per il 48 Kbyte si farà CLEAR 65311 poi si digiterà il programma, facendo attenzione ai DATA in quanto un eventuale errore potrebbe avere come conseguenza il blocco del calcolatore; completato e controllato l'inserimento, si darà il RUN.

A questo punto si può cancellare il caricatore e inserire il programma utilizzatore dei suoni.

Il loro uso è molto semplice, si richiamano con:

RANDOMIZEUSR 32544 e RANDOMIZEUSR 32574 per il 16 Kbyte  
RANDOMIZEUSR 65312 e RANDOMIZEUSR 65342 per il 48 Kbyte

la durata è contenuta in 65319 o in 65349 cioè +7 dall'iniziale;  
la frequenza è contenuta in 65317 o in 65347 cioè +5 dall'iniziale;  
il numero di ripetizioni è contenuto in 65313 o in 65343 cioè +1 dall'iniziale.

Attenzione a non scrivere degli zeri in quanto è facile che mandino in loop incontrollato il computer.

```
5 CLEAR 65311
10 FOR t=65312 TO 65367
20 READ a: POKE t,a
30 NEXT t
40 DATA 6,1,197,33,15,0,17,20,
3,223,205
50 DATA 181,3,225,17,16,0,167,
237,90,125,254
60 DATA 255,32,237,193,16,230,
201,0
70 DATA 6,1,197,33,0,3,17,2,0,
29,205
80 DATA 181,3,225,17,16,0,167,
237,82,32,240
90 DATA 193,16,233,201
```

Listato 1 - Versione per 48 Kbyte. Per la configurazione 16 Kbyte modificare le linee: 5 CLEAR 32543 e 10 FOR t = 32544 TO 32599.

«PER ACCORCIARE I TEMPI»

il numero di TELEX

del GRUPPO EDITORIALE  
JACKSON

è il seguente:

333436GEJITI

## Come controllare la memoria del TI99/4A

*Il signor Paolo Roncali di Cordenons (PN), propone alcune tecniche per tenere sotto controllo l'occupazione della memoria del TI99/4A ed evitare spiacevoli sorprese.*

Il seguente sottoprogramma può essere inserito all'inizio del programma principale oppure, cambiando i numeri di linea, chiamato al momento opportuno da una GO TO ...

```
1 KL = KL + 0.00883
2 PRINT KL
3 GO SUB 1
```

Dopo aver battuto il RUN, vedremo scorrere sullo schermo una continua sequenza di numeri che si fermerà solo alla comparsa del MEMORY FULL.

L'ultima cifra visualizzata corrisponderà appunto ai Kbyte di memoria che restano ancora liberi.

Il tempo impiegato è relativamente lungo: circa 20 secondi per Kbyte.

È possibile comunque diminuire notevolmente i tempi di attesa modificando il sottoprogramma nel seguente modo:

```
1 CALL CLEAR
2 PRINT "DOPO - MEMORY FULL - BATTI:":
"PRINT ::: KL (ENTER)"
3 KL = KL + 0.00883
4 GO SUB 3
```

Una volta raggiunto il MEMORY FULL bisognerà quindi battere PRINT ::: KL senza farlo precedere da un numero di linea; apparirà così un solo numero che corrisponde appunto ai Kbyte liberi.

Vediamo ora come è stato ricavato il numero 0.00883; ciò servirà soprattutto per adattare tale sottoprogramma ad altri tipi di calcolatori o comunque per altre capacità di memoria.

Per prima cosa è necessario eseguire (questa volta da solo) il seguente programma:

```
10 N = N + 1
20 PRINT N
30 GO SUB 10
```

Anche in questo caso vedremo scorrere sullo schermo una continua sequenza di numeri, che si fermerà solo alla comparsa del MEMORY FULL.

Nel caso del TI 99/4A l'ultima cifra è: 1811.

Si divide poi la capacità massima di memoria del computer (16 Kbyte) con l'ultima cifra letta (1811); il risultato di tale divisione è appunto 0.00883 che corrisponde al valore unitario di memoria.

Un'altra breve considerazione riguarda l'ultima istruzione del sotto programma: il GO SUB non può essere sostituito con GO TO!

Il primo sottoprogramma e di conseguenza anche il secondo, potrebbero essere modificati in diversi modi.

Per esempio:

```
1 KL = KL + 0.00883
2 KI = 16 - KL
3 PRINT KI, KL
4 GO SUB 1
```

fornirà sia la quantità di memoria impegnata dal programma principale (KI), sia quella libera (KL). Oppure:

```
1 KL = KL + 0.00883
2 NN = NN + 1
3 IST = 513 - INT(NN/3.5)
4 PRINT KL, IST
5 GO SUB 1
```

in questo caso avremo sia il valore della memoria ancora libera (KL) che il relativo numero massimo di istruzioni *medie* (IST) che sarebbe ancora possibile aggiungere al programma principale.

Naturalmente quest'ultimo valore non può che essere approssimato e deriva dalla considerazione che una media di 32 istruzioni occupa (nel TI99/4A) circa 1 Kbyte, purché tra queste non compaiano matrici da una o più dimensioni (una sola istruzione del tipo DIM Z(42,42) è sufficiente per avere il MEMORY FULL).

Il numero 513 che troviamo nella terza istruzione corrisponde al massimo numero di istruzioni *medie* possibili e lo si ricava moltiplicando 32 per 16 (Kbyte); mentre il numero 3.5 che figura nella mede-



## Come controllare la memoria del TI99/4A

sima istruzione è il risultato della divisione tra 1811 (già visto prima) e 513; rappresenta l'unità di istruzione *media* corrispondente all'unità di memoria (0.00883).

Vediamo ora un altro metodo, molto più semplice e soprattutto più rapido, anche se assai approssimativo, che permetterà comunque di sapere se è stato raggiunto oppure no un certo limite di memoria.

È sufficiente inserire (o se già esiste, aggiungere) nel programma principale il dimensionamento di una matrice fittizia, per esempio una DIM Z(450).

Avremo in questo modo una specie di lampada spia che indicherà la "riserva" nel "serbatoio" della memoria.

Infatti se dopo aver battuto il RUN non apparirà il messaggio MEMORY FULL avremo la certezza di avere ancora a disposizione oltre 4 Kbyte di memoria; nel caso contrario si può provare a cambiare il valore di dimensionamento della matrice, ad esempio con una DIM Z(228) ed in questo caso, se non comparirà il MEMORY FULL sapremo di avere ancora a disposizione più di 2 Kbyte.

Ritengo che la seguente tabellina sia sufficiente per una stima approssimata della memoria disponibile.

DIM Z(900)	8 Kbyte
DIM Z(450)	4 Kbyte
DIM Z(228)	2 Kbyte
DIM Z(114)	1 Kbyte
DIM Z(57)	0.5 Kbyte

Comunque volendo creare una tabellina più dettagliata si può ricorrere al seguente programma:

```

10 K = K + 0.00883
20 N = N + 1
30 PRINT N, K
40 GO SUB 10
  
```

Le cifre che si vedranno scorrere a sinistra dello schermo corrispondono al valore di dimensionamento della matrice e quelle a destra ai Kbyte.

Tale programma potrà quindi anche rivelarsi utile per conoscere quanta memoria occupa il dimensionamento di una matrice. ■

# EVM

post market

## PET-CBM

COMPILAT. DTL 4/8000	L. 330.000
COMP. PC/BASIC 3/4/8	L. 350.000
COMP. PET-SPEED 8000	L. 350.000
ULTRA KRAM	L. 300.000
SUPER BASIC	L. 90.000
DISK REVEALED	L. 60.000
LOCKSMITH	L. 40.000
WORD-CRAFT	L. 280.000
CHIAVI PROT. DTL DA	L. 75.000
EPROM PROGRAMMER PET	L. 270.000
80 COLONNE KIT DA 3032 A 8032	L. 260.000
CAVI DA	L. 50.000
APPLESOFT LOADER	L. 180.000
WORD PROCESS 3/4/8000	L. 180.000
ASSEMBLER 3/4/8000	L. 150.000

## VIC-20

40 COLONNE SOFTWARE	L. 45.000
40/80 COLONNE HARDWARE	L. 180.000
VIC EPROM PROGRAMMER	L. 180.000
VIC MON	L. 29.000
HIRES-3K GRAFICA	L. 54.000
MAGAZZINO	L. 150.000
RAM 3K	L. 49.000
RAM 8K	L. 75.000
RAM 16K	L. 118.000
PROGRAMMERS AID	L. 29.000
4 SLOT MOTHER BOARD	L. 69.000
FATTURAZIONE	L. 150.000

## CBM-64

BASIC 4.0 WEDGE	L. 95.000
SPRITE GENERATOR	L. 40.000
MAGAZZINO	L. 150.000
CASSA NEGOZI	L. 150.000
CAVO CBM64-CENTRONICS	L. 38.000
CHARACTERS GENERATORS	L. 30.000
FATTURAZIONE	L. 150.000
MUSIC	L. 40.000

## LIBRI

GUIDA AL VIC 20	L. 25.000
PERIFERICHE VIC	L. 9.500
CORSO DI ASSEMBLER CON PROGRAMMI SU NASTRO	L. 35.000
TAVOLE COMMODORE	L. 15.500
PERIFERICHE COMMODORE	in preparazione
GUIDA AL CBM 64	L. 25.000
IL BASIC 4.0 COMMODORE	L. 15.000
IL S.O. DEL 64 DISASS.	L. 38.000
CATALOGO EVM	gratis

## E.V.M. SOFTWARE BANK

Vi dà la possibilità di accedere e contribuire con pochissima spesa ad una vasta biblioteca di programmi. SCRIVETE PER INFORMAZIONI!!!!

Per ricevere il catalogo EVM POST MARKET su prodotti COMMODORE, SOFTWARE, PRODOTTI DI CONSUMO, COMPONENTI, STAMPANTI, ecc. **SCIVETECH!!!!** SCONTI AI SIGNORI RIVENDITORI.

### Condizioni di vendita:

I prezzi con la sola esclusione dei libri sono IVA esclusa. Franco Monteverchi. Pagamento Contrassegno.

IL NOSTRO INDIRIZZO È:

E.V.M. POST MARKET  
VIA MARCONI 9/A  
52025 MONTEVARCHI (AR)  
tel. (055) 982513

NOOME .....

INDIRIZZO .....

C.A.P. .... CITTÀ .....

### INVIATEMI

CATALOGO

INFORMAZIONI SU E.V.M. SOFTWARE BANK

.....

# PASCAL

## IMPARIAMO IL PASCAL

SCONTO 20% AGLI ABBONATI  
FINO AL 28-2-84

Compatezza, concisione, chiarezza e notevoli potenzialità scientifiche, oltre a prestarsi ottimamente per calcoli gestionali e ad essere usato anche con i microcomputer, sono le caratteristiche che decretano il successo del PASCAL come linguaggio di programmazione. Non vi era però finora un testo che insegnasse a tutti a programmare in PASCAL: o perché i libri esistenti sono troppo concisi, o troppo semplici, oppure perché richiedono la conoscenza di altri linguaggi di programmazione, o, non ultimo, perché in inglese.

Queste sono proprio le lacune che colma il libro un libro di divulgazione, incentrato sull'autoapprendimento, che non ledia con accademismi non funzionali il lettore riportandolo "a scuola". I capitoli sono il più possibile organici, in modo che la loro consultazione sia semplice ed agevole. Un riassunto di quanto si apprenderà è posto all'inizio e non in fondo al capitolo, perché il lettore possa subito avere un metro valutativo con cui verificare il suo apprendimento. E poi, ci sono consigli, problemi, esercizi affinché il libro sia "usato" e non letto, perché occorre sapere come si usa un'istruzione piuttosto che conoscerne le differenze semantiche tra linguaggio e linguaggio. Con un lavoro graduale, partendo senza alcuna conoscenza di programmazione, dopo circa due settimane dovrete conoscere abbastanza bene il PASCAL. Un buon risultato, no?!

# IMPARIAMO IL PASCAL

Pagine 162 Prezzo Lit. 11.500

EDIZIONE ITALIANA

FLAVIO WALDNER

GRUPPO EDITORIALE JACKSON



Per ordinare il volume utilizzate l'apposito tagliando d'ordine inserito in fondo alla rivista.

Formato 15 x 21

Codice 501A

### SOMMARIO

- |  |  |
|--|--|
| 0. Da non trascurare                           | 8. Gli statements logici                 |
| 1. Come si descrive la sintassi del linguaggio | 9. I dati strutturati - Generalità       |
| 2. Come si scrive in PASCAL                    | 10. Il tipo array                        |
| 3. Il programma e le dichiarazioni in generale | 11. Il tipo record                       |
| 4. Le dichiarazioni ed i tipi standard         | 12. Il tipo set                          |
| 5. I tipi speciali e subrange                  | 13. Il tipo file                         |
| 6. Gli statements di assegnazione              | 14. Il tipo pointer                      |
| 7. Gli statements di ripetizione               | 15. Le procedure e le funzioni           |
|  | 16. Procedure ricorrenti input ed output |
|  | 17. I diagrammi di struttura             |



GRUPPO EDITORIALE JACKSON

DIVISIONE LIBRI

# PICCOLI ANNUNCI

## Apple

**Compro** per Apple IIe 64 K programmi, Utilities, Games. Tratterei con appalista zona Milano per acquisto in blocco senza dovermi rivolgere. Spedire lista programmi e prezzi a Vittorio Gigante, Viale Casiraghi 491, 20099 Sesto S.G. (MI), tel. 2426171.

**Vendo computer** 48 K Plus minuscole e maiuscole totalmente compatibile Apple in contenitore identico all'Apple. Possibilità di collegarlo ad un normale TV b/n, L. 1.000.000, vero affare. Ora past. Paolo Anzolella, Corso Milano 81, 35100 Padova, tel. 049/98209.

**Cerco programmi Apple** corredati di manuali possibilmente italiani. Sono in possesso di Last One, DOS Tool Kit, Contabilità varia, ecc. con relativi manuali in italiano. Walter Franceschi, Via Binei 4/6, 11020 Donnas (AO), tel. 0125/82374.

**Vendo scheda espansione** 16 K RAM L. 100.000; scheda 80 colonne L. 120.000; scheda espansione 128 K RAM (DGS trasparente) L. 450.000. Tutto in ottimo stato per Apple II Plus. Isabella Rag. Bottini, Via G. Galilei 681, 18038 Sanremo (IM).

**Vendo-cambio programmi** per Apple II: Totocalcio (formule derivate, consecutivi, corr. di errore), Enalotto/Top (form. derivate, consecutivi, tazzine), Utilities, giochi, gestionali. Antonio Delisio, Via Tarquinia 7, 04100 Latina, tel. 0773/46300.

Quanti siamo con l'hobby del computer qui a Verona? Vi annunciamo in un club il mio P.C. è un Apple. Contattatemi se siete interessati! Marco Castellani, Via Ederle 7, 37023 Grezzana (VR), tel. 045/907998.

**Cambio-vendo programmi** per Apple con relativi manuali (quasi tutti in italiano). Inviatemi la vs. lista a cui farà seguito la mia. Vendo inoltre le riviste Informatica oggi n. 3/4/11/12 e P.S. n. 1/2. Milvio Mengarelli, Via Clementi 11, 20052 Monza (MI), tel. 328706.

**Vendo-cambio software** per Apple II. Dispongo di una vasta biblioteca di programmi. Rispondo a chiunque mi invia sua lista. Franco Vittor, Via Grabizio 35, 34170 Gorizia, tel. 0481/81254.

**Vendo Data Base personale.** Tool Kit a L. 50.000 l'uno (valore L. 150.000); in blocco L. 80.000, oppure cambio con giochi pari valore, compilatore Tasc o Visicalc. Diego Lagunas, Via Nicolo' Mauro 24, 31100 Treviso, tel. 0442/22898.

**Vendo Personal Computer MPF II.** 64 K RAM + 16 K ROM + Interfaccia per 2 minifloppy + Tastiera esterna + Minifloppy Slim-Line. Ancora in garanzia. Imballaggi originali. L. 2.100.000 trattabili. Giuseppe Pomponi, Via Raffaello 5, 56020 Castel del Bosco (Pisa), tel. 0571/467053.

A 2 mesi di vita **vendo computer** Orange costruito su commissione con tutti gli integrati su zoccolo (è identico a un Apple ma costa meno) + Monitor 12" forstori verdi + 3 manuali istruzioni in italiano + cassetta e alcuni programmi per L. 1.800.000 trattabili. Marco Pozzi, Via Mazzini 89, 50019 Sesto Fiorentino (FI), tel. 055/449293.

**Vendo solo in blocco** 50 dischetti software Apple per i vari contenuti grafici. Grafico Utilities, Gestionali, W.P. ecc. a L. 600.000. Richiedere elenco. Maurizio Degiannini, c. S. Santarosa 67, 12100 Cuneo, tel. 0174/61839.

**Cambio programmi e manuali** per Apple II di ogni tipo. Contattare ore serali o scrivere a Claudio Ogioni, Via Gonzales 4, 20139 Milano, tel. 5397867.

**Vendo** per Apple 2 i seguenti programmi: PFS File, Report, Graph, Visitrind, Visiplot, Visicalc, Apple Writer 2, Quick File, altri a 30% prezzo di listino + manuali e giochi ed Utilities in omaggio. Andrea Giacomobono, Via Torbole 6, 00135 Roma, tel. 30121782.

**Cambio e vendo programmi** per Apple II di qualsiasi genere; inviatemi i vostri cataloghi ed io risponderò con il mio! Marco Valente, via R.R. Perera 189, 00136 Roma.

**Cerco** drive usato per Apple purché in buone condizioni. Marco Mariani, Via Arenzano 74, 47023 Cesena (FO), tel. 0547/347514.

## Commodore

**VIC 20** a 4 libri + Joystick + C.A. 15 programmi + molti listati, il tutto in perfette condizioni, vendo per passaggio a sistema superiore a L. 510.000 trattabili. Giovanni Bono, Via Luosi 8, 20131 Milano, tel. 2367765.

**Per VIC 20 vendo** 3 Cartridges (Road-Race, Mole Attack, Sargon II Chess); sono stati comprati circa un mese fa e sono ancora in ottimo stato. Prezzo originali: Road-Race C.P. L. 490.000, Mole Attack C.P. L. 130.000. Possibilità vendita separata. Nicola Risimini, Via F. Condroni 8, 27100 Pavia, tel. 0382/303927.

**Vendo** per Commodore 64 cartuccia Hesmon da L. 70.000 conterranno oppure cambio con Hesmon per VIC 20. Scrivere a: Guido Rongoni, Via Graonetti 22/4, 33100 Udine.

**Vendo** miglior offerente Commodore 64 con floppy Disk, stampante, registratore cassette, istruzione e centinaia programmi. Telefonare (ore pasti): a: Gino Uglietti, Via Bandiera, 27058 Voghera (PV), tel. 0383/46932.

**Vendo Commodore 64** completo registratore L. 890.000; stampante per CBM 64 e VIC 20 L. 495.000; floppy disk CBM L. 595.000; dischetti, istruzioni e centinaia programmi L. 287.000. Telefonare ore pasti a: Gino Uglietti, Via Bandiera, 27058 Voghera (PV), tel. 0383/48932.

**Cerco possessori di Commodore 64** per scambio programmi e idee. Vendo listati di programmi originali inglesi per VIC 20. TI 99/44, Atari 400/800, Atom, Spectrum, ZX 81. Telefonare ore serali. Leonardo Fel, Via A. Fava 6, 20125 Milano, tel. 02/689412.

**Vendo-compro e scambio programmi** per Commodore 64. Scrivere o telefonare ore ufficio. Claudio Cambiè, Via Fatebenefratelli 26, 20075 Lodi (MI), tel. 0371/52402.

**Vendo programmi** per VIC 20 anche cassette da 20 programmi a 20.000 L. cad. Dispongo anche di programmi per 3K, 8K, 16K. Telefonare ore pasti: a: Marco, tel. 02/538191 o scrivere a: Marco Zanchi, Via Benaco 27, 20139 Milano.

**Vendo** per VIC 20 programma gestione contabilità semplificata su cassetta adatto per negozi. Luciano Usellini, Via Isonzo 31, Varese, tel. 0332/242556.

**Vendesi cartuccio VIC 20.** Slot, Poker cd L. 30.000; Joystick, miscelati cd L. 70.000; matematici, 1 e 2, biologia cd L. 40.000; stampante VC 1515 10 ore di funzionamento L. 350.000. Telefonare ore 20. Francesco Del Vecchio, Via Amoruso 34, 70124 Bari, tel. 510322.

**Vendo VIC 20 + C2N** registratore + 8K RAM + Programmers Aid + Mach. Language Monitor + Joystick. Miscelati cd del valore di listino di almeno 900.000 a L. 650.000. Carlo Galetti, Casella Postale n. 54, 01100 Viterbo, tel. 0761/229980.

Ho raccolto tutto quello che è stato pubblicato in Italia e all'estero sul C 64 (hardware e software). Il costo della raccolta fotocopiata è di L. 35.000. Scrivere per prenotazioni a: Francesco Del Vecchio, Via Amoruso 34, 70124 Bari, tel. 510322.

**Venditori programmi gestionali** per C 64. Condomini Arch clients, Word Processing, Mailing list, Utility, Grafica, Stampante grafica VC 1515 L. 350.000 10 ore funzionamento; cartuccia Hes Writer L. 50.000. Francesco Del Vecchio, Via Amoruso 34, 70124 Bari, tel. 510322.

**Vendesi L.** 10.000 dispensa tecnica sul DOS Manager e gestione dei Files relativi sul Floppy VC 1540 e 1541 per VIC 20 e C 64. Scrivere per prenotazioni a: Francesco Del Vecchio, Via Amoruso 34, 70124 Bari, tel. 510322.

**Cerco possessori di CBM 64** per scambio idee e programmi. Telefonare dalle ore 14 alle 15. Ivano Rossetti, Via Adda 9, 96100 Siracusa, tel. 60600.

**Vendo programmi (listati)** per computer Commodore VIC 20, 64; vendo inoltre programmi su ordinazione per Pocket Computer (max. 1,5 Kbytes) a L. 2.000 + 450 di spese di spedizione postale. Andrea Zanchetta, Via Dorso Durio 3488 L. 30123 Venezia, tel. 041/711043.

**Vendo VIC 20 + 16K RAM + Super Expander + registratore CN2 + 3 cassette giochi Jelly Monster, Omega Race, Rat Race + cassette (nastri) con programmi + manuali + alimentatore + libri vari, ore pasti.** Fabrizio Martini, Via Sargentina 8/2, 20092 Cinisello Balsamo (MI), tel. 6184717.

**Cerco disperatamente possessori milanesi di lettere** Floppy Disk L. 1541 per Commodore 64, per poter duplicare il dischetto test dei programmi di supporto. Ricambio con software musicale. Franco Francia, Via Cremonaso 6, 20148 Milano, tel. 02/392084.

**Per VIC 20 cambio-vendo software.** Possiedo circa 800 programmi, molti dei quali originali e in L.M.I.; inviatemi la vostra lista. Lionello Zanella, Viale Virgilio 21, 74025 Marina di Giosua (Taranto), tel. 099/627900.

**CBM 64** scambia, vende, compra programmi di vario genere possibilmente su disco. Richiedere lista. Desidero entrare in contatto con Commodore Clubs possibilmente zona (BO, RA, FE); assicurò risposta. Torricchio Giuseppe, Via Selice 42, 400216 Imola, tel. 0542/35409.

**Vendo VIC 20 + registratore** C2N + cartuccia Programmers Aid + manuali VIC Programmers Reference Guide e VIC 20 Revealed, il tutto a L. 450.000. Mario Oriandini, Via Murri 92, 40137 Bologna, tel. 349421.

**Vendo VIC 20 nuovo in garanzia** L. 350.000. G.C. Giacobbe, Via Finocchiarà 46, 16144 Genova, tel. 010/825537.

**Cerco utenti Commodore 64** per creare un club o scambiare esperienze solo nella zona di Perugia. Antonio Cherubini, Via Fonti Coperte 38/P, 06100 Perugia, tel. 075/33805.

**Vendo programmi di ingegneria civile** per PET/CBM. Telefonare negli orari di ufficio a: Gianluca Luoni, Via della Concordia 8/bis, 21052 Busto Arsizio (VA), tel. 0331/679128.

**Cerco programmi di ogni genere** su cassetta e cartidge per Commodore 64. Telefonare ore pasti allo 090/293821. Matteo Guido Vitetta, Via Ugo Bassi, isolato 157, 98100 Messina.

**Per Commodore 64 acquisto programmi** di ogni tipo (inviare lista per evitare doppi), riviste specializzate (soprattutto estere), routine. L'annuncio è sempre valido. Scrivere a: Marco Battagin, Via Satta 3, 20157 Milano.

**Per VIC 20** vendi i migliori programmi esistenti: originali inglesi, americani, linguaggio macchina. Programmi nuovi. Specifico catalano. Enzo Piperno, Via Torre dello Slinco 45, 00132 Fincocchio (Roma), tel. 6160934.

**Vendo (zona Palermo per verifica)** programma per VIC 20 che riduce sistema integrale tolocalcio. Il programma accetta qualsiasi condizione e massime comparazioni, col. base, segni consecut. ecc. Bruno Cacciatore, Via Calabria 4, lotto 43, 90144 Palermo, tel. 512302.

**Scambio software** CBM 3032 per ingegneria civile su cassetta o drive. Ing. Carmelo Micale, Piazza Nastasi 33, 96057 Milazzo (ME), tel. 090/924654.

Giovani utenti Commodore 64 cercano altri possessori zona Cosenza a provincia per scambio di informazioni e programmi ed eventualmente per fondare un club. Telefonare ore pasti. Vincenzo Garofalo, Via P. Colletta, 87100 Cosenza, tel. 0984/35710.

# PICCOLI ANNUNCI

**Compro per VIC 20 Super Expander, Programmer's Aid, VIC-MON, Joystick, Emilio Di Lello, Via Gioiò 3, 64026 Roseto d'Abruzzo (TE), tel. 085/8992146.**

VIC 20 usato pochissimo, come nuovo + interfaccia registratore cassette + modulatore per televisione + trasformatore 220 V + libro in dotazione + cassette con vari programmi, tutto L. 420.000. Per accordi telefonare ore pasti a: Franco Spadoni, Via Monte Priore 3, 60100 Ancona, tel. 06/2314452.

Vendo VIC 20 mai usato causa doppio regalo a L. 420.000. Offro in omaggio manuale in italiano e cassetta piena zeppa di giochi. Flavio Lucentini, Via G. Forzò 98, 00148 Roma, tel. 06/2331689.

Vendo programmi per Commodore, ingegneria civile ogni tipo di programma che possa essere utile per il calcolo strutturale + molti giochi (3032) esecuzioni programmi a richiesta; invio elenco principali programmi disponibili. Walter Ing. Cottafava, C.so Telesio 57, Torino, tel. 011/727965.

Vendo per VIC 20 cassetta contenente programmi in linguaggio macchina tra cui: Alien Blitz (versione base), The Alien 3K, The Tomb of Drowan 16K, Labirinto tridimensionale, 30 Mcz (v.b.), Othello (v.b.), Poker (v.b.) ed altri. Il prezzo è incredibile: L. 35.000. Paolo Anania, Via L. Capuana 56, 00137 Roma, tel. 823514.

Vendo VIC 20 + manuale in italiano + cassetta di costruttiva a sole L. 350.000 18 mesi di vita in ottime condizioni. Telefonare ore pasti, rispondi a: Marco Mussoni, Via Ferrara 28, 47037 Rimini (FO), tel. 08086.

Per cambio sistema cede Commodore 2001 (pietra 3032, Base 3000, Rom Tool Kit) espanso 32 K, registratore incorporato, completo di manuali e ricco software in cassetta (giochi, business, scientifico), vera occasione, tutto a L. 750.000 trattabili. Telefonare ore pasti. Massimo Vittor, Via Grabazio 35, 34170 Gorizia, tel. 0481/81254.

VIC 20 espansione 8 K ottimo stato vendo L. 450.000 solo Viterbo e provincia. Franco Quintini, Via Dalmazia 84, 01100 Viterbo, tel. 39472.

Si vendono o si cambiano programmi per Commodore serie 3000-4000; per ricevere altro elenco inviare L. 1.000. Si producono anche programmi su commissione. Scrivere o telefonare ore pasti. Aroldo Bizzari, Via Pantelleria 19, 91100 Trapani, tel. 0923/20044.

Vendo per VIC 20 o scambio: Metecuron, Alien, Suplerlander, Starblitz, e su cassetta: Ladders, Serpente, Sottomarino, Othello ed altri giochi e utilità, più di 100 programmi. Telefonare o scrivere a: Fabio Zaccaria, Via dei Colli 6, 00020 Sambuci (Roma), tel. 0774/7108.

VIC 20 scambio programmi, listato a richiesta; acquistare Vicstat e Vicgraf con istruzioni. Fernando Forner, Via Valperga Caluso 21, 10125 Torino, tel. 011/6506538.

Vendo VIC 20 + Drive 1540 + Printer 1515 + 16 K, Vicstat, Progr. Aid, Graphic, Assembler + cav. alimentatore, registratore, dischi, manuali, val. acquisto 2,5 milioni, miglior offerente base 1,8 milioni. Franco Braga, Via L. Pasteur 18, 24047 Treviglio (BG), tel. 0363/49221.

Vendo per VIC 20 (3K - 8K) e Pet 2001 Word Processing Utilities vera e bellissimi giochi, in cassetta o su disco; a richiesta si invia ampia documentazione. Scrivere o telefonare allo 0372/27498, Carlo Zanini, Largo Paolo Sarpi 17, 26100 Cremona.

Vendo Floppy Disc x VIC 20 e 64, garanzia da convalidare, causa motivi finanziari. L. 600.000. Amos Aimi, Via Zanella 11, 43015 Noceto (Parma).

Vendo cassetta giochi VIC 20 a L. 30.000. Tra i più famosi troviamo: Alien Blitz (3,5 K RAM), Money Money (3,5 K RAM), Labirinto 3D (8 K RAM), Alberto Frabetti, Via De Nicola 66, 40100 Bologna, tel. 051/664285.

Vendo schema elettronico VIC 20 suddiviso in 3 parti più strutture interne dei seguenti integrati: Processore 6502, VIC 6522, VIC 6561, Buffer 6525, Demultiplexer 74 LS 158, int. bilaterale per la trasmissione dei segnali digitali 4066 e tanti altri schemi correati di tavolo della verità. Il tutto a L. 10.000. Vendo anche gli schemi separatamente. Massimiliano Ugolini, Via L. Magi 34, 53041 Asciano (Siena), tel. 0577/18398.

Cambio e vendo giochi per Pet 2001-3032 da L. 5.000 a L. 15.000. Telefonare la mattina. Fabio Talamini, Via Eroi del Lavoro 12, 04100 Latina, tel. 498036.

Vendo per VIC 20 oltre 300 programmi di ogni genere a prezzi vantaggiosi; per ricevere le liste inviare L. 400 in francobollo. Loredana Raponi, Via Alberto da Giussano 5, 00176 Roma, tel. 2754205.

Vendo per VIC 20 (3,5 Kb) 1 cassetta con 14 videogames L. 20.000, 1 cassetta con 7 programmi matematici L. 15.000, 1 cassetta con 8 programmi (Memory/Map, grafica, punto punto ecc.) L. 15.000, 1 cassetta con gestionali L. 10.000. Marco Citerio, Via Balassi 5, 22043 Galbiate, tel. 0341/540358.

Vendo Commodore 3032 + Floppy 2031 + stampante Honeywell + box sonoro + riviste Bit ed altre + due libri per il Pet + dischi (37) pieni di programmi + 4 nuovi + libri relativi ad uso programmi ed altro a L. 4.000.000. Andrea Chiappi, Via Dicozzano 41, 33010 Feletto Umberto (UD), tel. 0432/681479.

Compro programmi di ogni genere, listati o su cassetta per VIC 20. Massimo Tabasso, Piazza Molteni 1, 12038 Savigliano.

Scambio programmi per VIC 20; invio cassetta con programmi a chi mi invia cassetta con programmi; scambio listati per VIC 20, risponde a tutti. Marco Ghirardelli, Via Caduti per la Libertà 30, 48015 Cervia (RA), tel. 0544/971697.

VIC 20 vendo giochi in linguaggio macchina a prezzi modici ed altri programmi interessanti (compilatore, assembler). Cesi Marcello, Via Magliana nuova 178, sc. B, int. 22, 00146 Roma, tel. 06/5266009.

Vendo cassetta giochi VIC 20 L. 30.000. Tra i più famosi troviamo: Alien Blitz 3,5 K, Money Money 3,5 K, Labirinto 3D 8K, ecc. Alberto Frabetti, Via E. De Nicola 66, 40132 Bologna, tel. 051/664285.

Vendo videogiochi per VIC 20 inespanso e non, su cassetta. Nessuna protezione sul software e documentazione ai programmi. Per listino inviare L. 1.000. Telefonate a: Roberto Salerno, Via 4 Novembre 24, 28068 Romeno (NO), tel. 0321/200000.

Cambio-vendo programmi di vario genere per VIC 20 a chiunque mi richieda l'elenco glielo manderò gratuitamente. Chi mi manderà la sua cassetta io gli offrirò 10 programmi in più di quelli richiesti. Roberto Ossellatore, Via Fausta 136/A-5, 30010 Ca' Saverio (VE), tel. 966923.

Vendo-cambio per VIC 20 software di alta qualità a bassi prezzi! Scrivete a: Carlo Borso, Via G. Baris 34, 18100 Imperia, o meglio ancora telefonate 0183/21833.

Vendo Pet 2001 Commodore con registratore incorporato completo di manuali e di richiamo software in cassetta, tutto in ottime condizioni, a L. 800.000 trattabili. Vera occasione. Per informazioni telefonare ore serate o chiedere di Franco o lasciare indirizzo. Franco Vittor, Via Grabazio 35, 34170 Gorizia, tel. 0481/81254.

Cerco possessori VIC 20 in Romagna per scambio idee, esperienze, programmi. Contattarli anche Vic Club. Alessandro Capasso, c/o Brigliadori, Via D. Samory 38, 47100 Forlì.

Vendo, cambio e cambio software per Commodore 64. Inoltre cerco altri utilizzatori (zona Trieste) per eventuale creazione di un club. Scrivere o inviare liste programmi a: Roby Delbello, Via Gianzoleto 7, 34100 Trieste.

Vendo VIC 20 + 3 K RAM Super Expander + Programmer's Aid + Sargon II Chess + manuale + libro sul 6502, il tutto 3 mesi di vita; regalo programmi. Massimo Gaspari, Via Amarena 5/5A, 16143 Genova, tel. 010/875264.

Vendo listati di 1600 programmi per VIC 20 a prezzo modico; speditemi richieste + francobollo per risposta. Paolo Perlini, Via 271 Gennaio 152, 61020 Montecchia (PS), tel. 072/919360.

Vendo VIC 20 + registratore C2N + manuale in italiano + cassetta Cartridge "scacchi" a L. 550.000 non trattabili. Telefonare ore pasti. Carlo Cosentino, Via Marco Attilio 14, 00136 Roma, tel. 319224.

Vendo software-hardware interessante per VIC 20 anche espanso. Massima serietà. Scrivere a: Giuseppe Ziggio, Via Lungo Chiampio, 36054 Montebello (VI).

Vendo VIC 20 usato pochissimo, con Joystick, 6 cartucce giochi + Programmer's Aid, unità cassette C2N e libri sul VIC 20. Prezzo trattabile. Telefonare ore pasti. Daniele Dal Bello, Viale Firenze 77, 06034 Foligno (PG), tel. 0742/20103.

Possessore da poco di VIC 20 cerca users di questo computer per scambio idee e programmi. Michele Massa, Viale Ponte dell'Armadorio 66, 16148 Genova, tel. 336605.

Vendo VIC 20 + cartuccia gioco Mole Attack, il tutto appena acquistato (1 mese di vita) a L. 450.000. Solo zona Rimini. Luigi Barberini, Via Mengo L. 70.000, Rimini (FO), tel. 0541/772021.

Vendo-cambio VIC software Basic e L.M. con o senza espansione. Giorgio Pietrocchia, Via Canova 25, 01159 Roma, tel. 4388795.

CBM qualsiasi configurazione vendo programma presentazione effetti sconto/SBF con stampa distinta e calcolo compenso. Scrivere o telefonare ore pasti. Vincenzo Santarca, Via Res. Partigiana 116, 97015 Modica (RG), tel. 0932/300538.

Cerco programmi per contabilità generale e fiscale ordinaria e semplificata listati o cassetta per VIC 20. Mario Cima, Via Gioiò 70, 80128 Napoli, tel. 081/378665 ore pasti.

Vendo VIC 20 + registratore 2CN + superespansione (8K + grafica) + Bridge gioco del poker + manuali e tutto il software fatto su cassetta a L. 600.000. Stefano Albanesi, Via Leopardi 10, 20123 Milano, tel. 8055804.

Per VIC 20 vendo: Light-Pen L. 35.000, 16K RAM L. 12.000, HI-RES L. 30.000, Vicon L. 30.000, Tool Kit L. 20.000, Rotmer Board (7 Slot) L. 100.000, 8K RAM L. 70.000, Nicola Pedrol, Via Vigevanese 36, 20010 Boregno (MI), tel. 9013314.

Cerco possessori possibilmente in zona Padova di un Commodore CBM 64 per scambio software e idee. Desidererei sapere se vi sono attualmente in Veneto club di utenti Commodore 64. Giovanni Borghi, Via Pascoli 1/int. 1, 35100 Padova, tel. 049/692418.

Vendo VIC 20 come nuovo con alimentatore + modulatore TV + espansione 3K RAM + interfaccia + registratore + manuali originali + numerosi programmi anche in L.M. Prezzo veramente eccezionale. Marco Mancinelli, Via Chiaravalle 3, 60127 Ancona, tel. 898616.

Vendo software Commodore 64, programmi scientifici, giochi vari, dettagliata documentazione. A richiesta invio catalogo gratuitamente. Risponde a tutti. Luciano Antonucci, Via C. Goldoni 7, 05100 Terni, tel. 0744/21274.

Vendo per VIC 20 cartucce Super Expander + 3 K, Tool Kit, Monitor, ing. macchina, 16 K, K. Slot, Jupter Lander, Alien, 50 di scacchi, Jupter Vendo Cabinet 7 schede Arco con coprerchio e alimentatore L. 200.000. Ferdinando Cosentino, C.so Europa 16, 83100 Avellino, tel. 0825/22909.

# PICCOLI ANNUNCI

**Cerco programmi di ogni genere listati o cassetta per Commodore 64. Telefonare ore serali. Pietro Placchi, Via F.lli Rosselli 22, 48018 Faenza (RA), tel. 0546/30374.**

**Vendo VIC 20 + C2N + 8 K + Vicgraph + Tapes Vicheck, Vicalc, Vicat, tutti del gennaio 1983, perfetto stato, manuali italiani, imballo originale a L. 700.000 (val. reale ca. 975.000). Telefonare o passi a scrivere a: Sandro Salvati, Via G. Zoccoli 53, 41040 Baggiovara (MO), tel. 059/359708.**

**Comprò Cartidge (cartuccia) per VIC 20 "Jelly Monsters" nuova o usata, solo se in buono stato e funzionante. Prezzo da concordare. Lucia Macconi, Via Pismonte 5, 20139 Milano, tel. 02/5391991.**

**Vendo VIC 20 + molti programmi + Superexchanger + Vicmon + registratore + 2 libri e Reference guide. Il tutto a L. 500.000 solo zona Milano. Fulvio Tonello, Viale Francesco Crispi 5, 20121 Milano, tel. 02/653092.**

**Vendo VIC 20 + 16 K + C2N + Joystick + cassetta dimostrazione usato solo 2 mesi per cessata attività a L. 650.000. Telefonare ore pasti. Paolo Decarli, Via P.F. Calvi 12, 39100 Bolzano, tel. 0471/25149.**

**Vendo programmi per VIC 20 su cassetta. Richiedere elenco inviando francobollo o propria lista. Alessandro Comisso, Via Montebianco 12, 20090 Cesano Boscone.**

**Scambio software per Pet Apple, paghe, ingegneria, contab. gestionale, magazzino, fatti, radiocamtori RITTY/RX, TX, Mailbox, CW, o scambio con RXTX per decamatrice, funzionante. Ore 20,30, IW6MEQ, Stella Paola, Via Nicola Moscardelli 28, 67100 L'Aquila, tel. 23273.**

**Vendo per VIC 20 Joystick L. 10.000, VIC Forth L. 80.000, cassetta programmi vari L. 10.000 + 1 sequenzi giochi su cassetta VIC Scramble L. 1.000, Amok L. 15.000. Telefonare dopo le 15 solo zona Milano. Diego Zappa, Via Cairoli 46, 20017 Rho (MI), tel. 0397251.**

**Per Commodore 64 compro-scambio software solo su cassetta. Eventuale scambio di esperienze ed idee. Telefonare dopo le 20,30, Paolo De Luca, Via De Vesiani 36, 35100 Padova, tel. 049/755745.**

**Per Commodore 64: Vendo Utility o Games originali inglesi su cassetta o dischetto. Scrivere per richiesta lista e prezzi a: Stefano Sfriglia, Viale Vittorio Emanuele 47, 21400 Bergamo.**

**Vendo per Commodore 64 cartuccia Turtle Graphics II come nuova con libretto istruzioni a L. 95.000. Permette la grafica HI-RES con le istruzioni del Logo, Mario Pelliccioni, Largo Appio Claudio 385, 00174 Roma, tel. 7484916.**

**Cerco listati di ogni genere progr Commodore 64. Scrivete o telefonatemi: Simona Solari, Via Baisi 1/4, 16035 Rapallo (GE), tel. 0185/51223.**

**Vendo VIC 20 + unità cassetta C2N + espansione 8 K RAM + manuale italiano-inglese a cinque giochi (Jelly Monsters, Rat Race, Super Lander, Scacchi, Amok), tutto a L. 600.000. Daniele Colonna, Via Savoldo 3, 20125 Milano, tel. 6425169.**

**Vendo progr. su cassetta per VIC 20 a prezzi bassissimi; ampia scelta circa 500 progr. (molti in L.M.). Scambio sempre per VIC solo progr. in L.M. Scambio anche progr. per ZX Spectrum. Allegare 700 L. per le liste. Gratuite se inviate le vostre. Giuseppe Mascali, Via R. Margherita 573, 98028 S. Teresa Riva (ME), tel. 0942/791692.**

**Cerchiamo VIC ing. e possessori di MPF II. Saremo incentivati a formare club dei citati computers su Treviso. Disponiamo di molti programmi. Rispondiamo a tutti. Andrea Giordani, Via Montagna Grande 11, Montebelluna (TV), tel. 0423/20301.**

**Vendo VIC 20 programmi di vario genere su nastro di cui alcuni in linguaggio macchina. Telefonare dopo le ore 17,30 allo 02/2421130 oppure scrivere a Paolo Lambri, Via Alfieri 60, 20099 Sesto S. Giovanni (MI).**

**Cerco per Pet 4032 programmatore di Eprom, scambio con prog. contabilità magazz. IVA, fatt. ingegner., radiocamtori (RITTY, CW, Mailbox), giochi, Utility, Source processor. Telefonare allo 0662/23773. Paolo Stella, Via Nicola Moscardelli 28, 67100 L'Aquila.**

**Vendo VIC 20 usato pochissimo + scheda alla risoluzione + interfaccia per qualunque registratore + manuale in italiano + numerosi programmi. Tutto in blocco a L. 450.000. Giorgio Bertolina, Via Miniere 6, 10015 Ivrea (TO), tel. 0126/423759.**

**Cerco possessori di Commodore 64 per scambio software. Informazioni varie. Inviare le vostre liste a Sordini Roberto, Viale Kennedy 27, 61032 Fano (PS), tel. 0721/873152.**

**Vendo per VIC 20 e Commodore 64 3 giochi di avventura originali americani. Non richiedono l'espansione di memoria e costano L. 25.000 spese di spedizione comprese. Alberto Frabetti, Via E. De Nicola 66, 40100 Bologna, tel. 051/564285.**

## Olivetti

**Vendo per Olivetti M20 programma professionale fotocopia. Sviluppo e stampa su schede sistemi integrali, derivati, correzione errore. Ridotti fino a un milione colonne. A L. 500.000, più di 32 K! Davide Agnola, Via Monte Suello 19R, 16129 Genova.**

**Verifica occasione. Vendo stampante termica Olivetti Ope TH 240 (240 linee 80 caratteri) usata pochissimo a L. 500.000 + interfaccia per Spectrum a L. 50.000. Francesco Giorgio, Via Circonvallazione 42, 00018 Favone Canavese (TO), tel. 0125/51325.**

**Cerco libri e manuali che mi insegnino a programmare e a conoscere tutto sul computer Olivetti BCS 2035. Sono interessato anche a programmi applicativi di contabilità banche. Sergio Di Giorgio, Via Comunale Napoli 99, Int. 16/bis, 80126 Pianura (NA), tel. 7266184.**

## Sharp

**Vendesi Sharp 1500 + stampante 4 colori CE 150 (espansione 5 K + 150.000 compres. manuali; registratore digitale Philips D6600 L. 80.000. Telefonare ore 20. Francesco Del Vecchio, Via Amoruso 34, 70124 Bari, tel. 510322.**

**Per Sharp MZ 80 K vendo bellissimi giochi di animazione, lista di indirizzi di utilissimi Poke, implementazioni del Basic, Pascal, System Programs, Word Processing, e molto altro. Claudio Giovanelli, Via Ripamonti 194, 20141 Milano, tel. 02/536926.**

**Vendo-cambio programmi per Sharp MZ 80B (ingegneria civile, Word Processing, giochi, pronostici localoc). Scrivere per la lista o per ulteriori informazioni. Cerco solo per fondare club. Stefano Lazzarini, Via Monte Sabotino 2, 35141 Padova, tel. 049/22675.**

**Vendo Sharp MZ 80K 48 Kbite con manuale, Super Basic, Sharp a 15 cifre, interfaccia x Epson 100, tastiera numerica L. 2.000.000 trattabili + L. 1.600.000 x stampante non obbligatoria. Bruno Montaguti, Via Soli 1, 41058 Vigonza (MO), tel. 059/771116 cire negozio.**

**Per Sharp MZ 80 K vendo bellissimi giochi di animazione, lista di indirizzi di utilissimi Poke, implementazioni del Basic, linguaggi Pascal, System Programs, Renumber-Append, Word Processing ed altro. Claudio Giovanelli, Via Ripamonti 194, 20141 Milano, tel. 02/536926.**

**Vendo Sharp PC 1500 completo di manuali in italiano e accessori a L. 4.750.000. Offro in omaggio diversi programmi originali. Telefonare ore pasti 0584/96603. Roberto Girola, Via Dei Memi 12, 55041 Piano di Conca, Viareggio (LU).**

**Vendo Sharp PC 1211 + CE 122, perfetto stato, manuali e programmi vari in italiano, imballo originale a L. 300.000. Telefonare ore pasti o scrivere a: Sandro Salvati, Via G. Zoccoli 53, 41040 Baggiovara (MO), tel. 059/359708.**

**Vendo listato per PC 1211 per sviluppo integrale sistemi fotocopia da 1 a 13 triple L. 30.000. Cosimo Napolitano, Viale Comandini 28, 72100 Brindisi, tel. 0831/84170.**

## Texas

**Vendo Texas TI 99/4A usato 2 mesi garanzia in bianco con Joystick K tutto a L. 400.000. Telefonare ore pasti a: Alessandro De Chiara, Via Niccolini 10, 50121 Firenze, tel. 055/663622.**

**Cerco possessori di TI 99 Texas per scambio e acquisto programmi. Ho 26 anni ma è da poco più di 3 anni che ho un Home Computer. Telefonatemi solo dalle 20 alle 22 allo 02/707780. Milano, Giordano Ambrosetti, Via F. Bellotti 7, 20129 Milano.**

**Vendo software su modulo (ROM) per poter programmare in Basic esteso con il Texas TI 99/4A. Prezzo interessante. Telefonare 049/811213 o scrivere a: Maunilo Bregolin, Via G. Puccini 4, 35031 Abano Terme (PD).**

**Scambio programmi per TI 99/4A in TI Basic e cerco appassionato disposto a scambio di informazioni e possibilmente manuali riguardanti il linguaggio Assembler. Telefonare ore pasti (meglio zona AL). Massimo Simeoni, Viale Vicenza 3, 15048 Valenza (AL), tel. 0131/957178.**

**Vendo TI 99/4A Computer + cavi interfaccia X2 registratori + cassetta programmi + 2 manuali e varie documentazioni tutto a sole L. 490.000. Roberto Falchi, Via Puglie 22, 09013 Carbonia (CA), tel. 0781/67124.**

**Vendo programmi di giochi e no per TI 99/4A. Alcuni sono originali TI. Sono disponibili anche programmi per il TI Extended Basic. Richiedere catalogo per lettera a: Andrea Barbieri, Via Livorno 12A, 35100 Padova.**

**Vendo programmi in Basic per il TI 99/4A. Inviare un francobollo per ricevere il listino prezzi. (Molti giochi e programmi gestionali ottimi prezzi). Luca Osella, Corso Chiari 707, 10132 Torino, tel. 011/897054.**

**Vendo programmi per il TI 99/4A anche in Extended Basic. Ecco alcuni titoli: Hamurabi, Maricet Simulation, Helicopter e tanti altri. Richiedere lista con prezzi a: Andrea Barbieri, Via Livorno 12A, 35100 Padova, tel. 658995.**

**Vendo Joystick per TI 99, VIC 20, Atari da competizione a L. 35.000; programmi giochi super per TI 99 su cassette, 5 per L. 10.000, 15 per L. 20.000. Anticipato o contrassegno. Giuseppe Barca, Via Tre He 29, 20047 Brugherio (MI), tel. 039/879211.**

**Vendo TI 99/4A in garanzia completo di cavo registratore + cassetta "Basic per principianti" ed altri programmi a sole L. 290.000. Giampiero Gallina, Via Carrozzeri alla Posta 13, 80134 Napoli, tel. 313175.**

**Vendo programma poker per Texas TI 99/4A ottima grafica per 1 o più giocatori, simile al gioco dei bar: cassetta L. 25.000, disco L. 30.000 comprese spese. Programmi, mediante vaglia postale. Giuseppe Persiano, Via Alfredo Capone 36, 84100 Salerno, tel. 089/350734.**

**Texas TI 99/4A Users Unitver! Per creazione Computer Club Turinese scrivete o telefonate a Franco Rolandi, Via S. Trassoni 47, 10143 Torino, tel. 011/766724.**

**Vendo Texas TI 99/4A con accessori, manuale e interfaccia per 2 registratori, garanzia databile, perfetto condizioni, per passaggio a sistema superiore. Il tutto imballato per sole L. 310.000. Dario Riganotti, Via Lombardia 51, 20036 Meda (MI), tel. 0362/73041.**



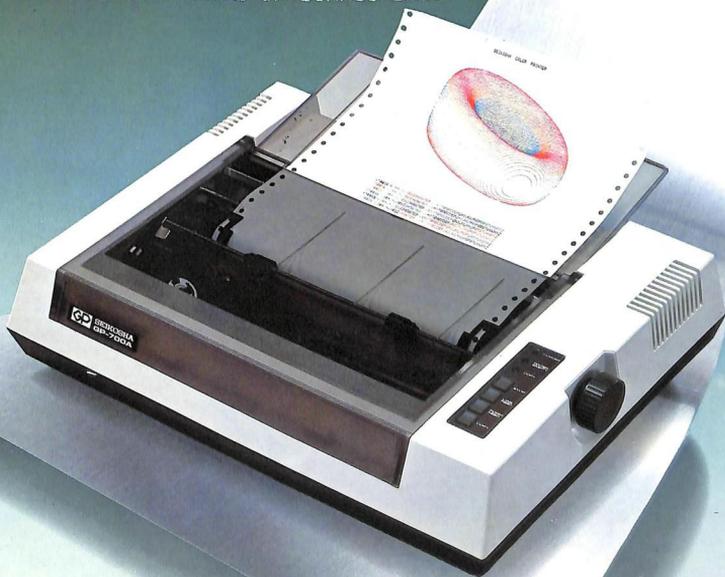
# QUANTI COLORI HA LA TUA STAMPANTE ?

NEL 1983 LA SEIKOSHA PER PRIMA AL MONDO  
E' IN GRADO DI PRESENTARE LA NUOVA STAMPANTE  
GRAFICA A SETTE COLORI.

RIUNITE IN UN APPARECCHIO PRATICO E COMPATTO  
LE CARATTERISTICHE DELLA STAMPANTE E DEL PLOTTER.  
LA SEIKOSHA INVENTA UN NUOVO TIPO DI PERIFERICA  
CHE BEN PRESTO SARA' INSOSTITUIBILE.

REBIT COMPUTER E' ORGOGLIOSA DI LANCIARE  
QUESTA NOVITA' ASSOLUTA SUL MERCATO ITALIANO  
AD UN PREZZO MOLTO, MOLTO COMPETITIVO:  
MENO DI UN MILIONE.  
MENO DI UNA COMUNE STAMPANTE IN BIANCONERO.

**REBIT**  
COMPUTER  
A DIVISION OF G.B.C.



## GP-700A

**Graphic Color Printer**

**SEIKOSHA**



# ce l'hai?

Il tuo Spectrum è preziosissimo difendilo con la "SUPER GARANZIA"  
La Rebit Computer, distributore per l'Italia dei prodotti SINCLAIR, ha messo a punto la nuova straordinaria

## SUPER GARANZIA

Apri la scatola del tuo SPECTRUM acquistato presso un Rivenditore Autorizzato e ci trovi anche un libretto: ti accompagnerà nei tuoi futuri acquisti, dandoti l'occasione per risparmiare oltre 100.000 lire. Ti darà la Garanzia di una perfetta assistenza, e avrai la certezza del valore del tuo autentico SPECTRUM. Il libretto della "SUPER GARANZIA" contiene le modalità per l'iscrizione al SINCLUB, la federazione di tutti i Sinclair Club Italiani. Inoltre il Coupon sconto per abbonarsi a "SPERIMENTARE" il mensile di elettronica che pubblica il bollettino Sinclub: idee, programmi, notizie, vita associativa.

La tessera Software ti dà diritto ad uno sconto sull'acquisto dei programmi. Infine nel libretto "SUPER GARANZIA" troverai la possibilità di acquistare la stampante ZX PRINTER SINCLAIR ad un prezzo eccezionale.

PER QUESTO UNO SPECTRUM  
SENZA LA "SUPER GARANZIA"  
E' SOLO UN MEZZO  
Spectrum



# sinclair

## Spectrum

molto di più di una garanzia!!

**REBIT**  
COMPUTER

A DIVISION OF G.B.C.