

Sinclair COMPUTER

mensile per gli utenti dei computer Sinclair

#12 Lire 3000
mar/apr 1985

Un completo
disassemblatore
per lo Spectrum

speciale!
.....
accesso random
ai microdrives



Systems



COMPUTER SHOW

85

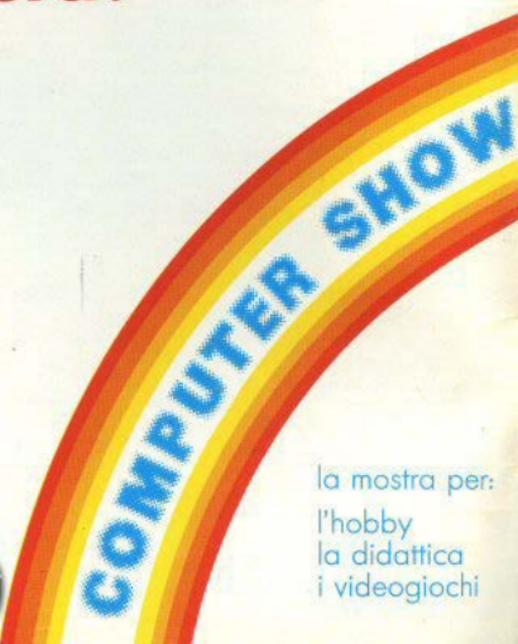
19 - 23 APRILE 1985 -  FIERA DI MILANO

Ogni anno circa due milioni di persone visitano la Fiera Campionaria di Milano. Dal 1985, ad attenderle, ci sarà **COMPUTER SHOW**

il nuovo Salone interamente dedicato all'informatica per i giovani, la scuola, la famiglia moderna. Sicuramente sarà il più grande appuntamento dell'anno.

Perché non esserci?

Segreteria:
E.P.I. - ENTE PROMOZIONE INFORMATICA
Via Marochetti, 27 - 20139 Milano
Telefoni (02) 56.93.973 - 53.98.267



COMPUTER SHOW

la mostra per:
l'hobby
la didattica
i videogiochi

sommario

- 04 - Sinclairamente vostro/la posta
- 10 - Libri/recensioni
(Maubart)
- 12 - Servizi: Accesso casuale ai microdrives
(R. Zonin/D. Bolla)
- 16 - Software: DISASSEMBLATORE
(Giovanni Mellina)
- 18 - Hardware: EPROM/II: L'edge connector
(Sergio Coraglia)
- 20 - Computer music: MUSIC COMPILER/II
(Guido Grassi)
- 21 - Didattica: ASSEMBLY
(Gianluca Carri)
- 22 - Software: AutoDATA (G. Mellina)
Joystick (Raffaele Ilardo)
Billy (Luca Ghisleni)
Battaglia sottomarina (Ciaspe)
- 23 - Software: TOTOSeikoshia
(Fabio Berno)
- 25 - Software: GRAFICI
(Marco Manunta)
- 26 - I listati
- 45 - Didattica: Le variabili di sistema
della ROM-ombra
- 46 - Software: Orologio digitale
(Stefano Furlani)
- 47 - Software: Checksum per il basic
(Giovanni Mellina)
Sillabe (Luca Ghisleni)
- 48 - I risultati del benchmark del lettore
(Paolo Beneventi)
- 53 - Didattica: Un tasto per volta
(Maubart)
- 58 - Videogames/ recensioni
(Luigi Callegari)
- 60 - Sinclairreclame/piccoli annunci

in copertina: QL Chess, gli scacchi della
PSION per il Sinclair QL

sinclair COMPUTER

REDAZIONE
Mauro Soldavini, Fabio Berno,
Marco De Martino

SEGRETARIA DI REDAZIONE
Maura Ceccaroli, Piera Perin

COLLABORATORI
Paolo Beneventi, Marco Bertani, Damiano Bolla, Giuliano Boschi, Luigi Callegari, Gianluca Cieri, Valerio Cipolla, Paolo Dray, Fabrizio Ferraro, Monica Fumagalli, Stefano Furlani, Guido Grassi, Giovanni Mellina, Antonio Russo, Renzo Zonin.

GRAFICA E IMPAGINAZIONE

Cristiana Oglio
FOTOGRAFIA
Ferruccio Mustilli

DIFFUSIONE E ABBONAMENTI
Marina Vantini

DIREZIONE, REDAZIONE
Viale Farnagosta 75 - 20142 Milano
Tel. (02) 8467346/9/40

PUBBLICITÀ
Milano: Mirco Croce (coordinatore), Giuseppe Porzani, Michela Prandini, Giorgio Ruffoni, Claudio Tidone, Villa Claudio
Segretaria: Liliana De Giorgio
V.le Farnagosta 75, 20142 Milano - tel. (02) 8467346/9/40
Roma: SpazioNuovo di R. De Marinis via P. Foscarini 70, 00139 Roma tel. (06) 8109679

Torino: Spaziobelli di Daniele M. Costagna - via Piazzetta 50, 10134 Torino - tel. (011) 327617

FOTOCOPOSIZIONE
Fotocomposizione LM (Brescia)

STAMPA
La Litografica S.r.l. (Busto Arsizio)

DISTRIBUZIONE
Messaggeria Periodici S.p.A.
via G. Carcano 32, Milano
Spedizione in abb. Post. GR. III/70
Direttore responsabile:
Agostina Ronchetti
Autoriz. Trib. di Mi n. 255/12.11.1983

Una copia L. 3.000 (Arretrati L. 8.000)
Abbonamento annuo (11 numeri) L. 28.000 (estero il doppio). I versamenti e le richieste di arretrati vanno indirizzati a: Sinclair Computer, V.le Farnagosta 75, 20142 Milano, mediante emissione di assegno bancario o versamento sul c/c postale n. 37952207 intestato a SYSTEMS editoriale. Per i cambi di indirizzo indicare, unitamente al nuovo, anche l'indirizzo precedente, allegando L. 500 in francobolli.

Sinclair ZX81, ZX Spectrum, ZX Microdrive, QL sono marchi registrati della Sinclair Research Ltd.





sinclair *amente
vostro*

Proteste

Sono un abbonato alla vostra rivista Sinclair Computer fin dal n. 2 perchè nonostante gli inizi scadenti, ho voluto dare fiducia e aiuto a questa pubblicazione portata verso un unico indirizzo: gli elaboratori Sinclair, che prediligio dalle ZX80. Così, perseguendo nella mia testardaggine, ho acquistato anche il n. 1 della rivista su cassetta 16/48, edito da voi per la versione italiana, mia seconda esperienza su tali pubblicazioni.

A parte l'abissale diversità di contenuto editoriale, dovuto alla differenza di prezzo (5.800 contro 9.000), (ma da che cosa? - NdR), la cosa che più mi ha fatto soffrire è stato il cercare di caricare i programmi, quasi tutti al limite dell'intelligibilità sonora, problema risolto solo per 10 programmi su 12 e grazie ad un'ampia collezione di registratori. Infatti rumori di fondo, truscii e chissà cos'altro, abbondavano. E fosse finita qui...

Tul n. 9 di SC, del dicembre '84 e giuntomi solo oggi e non per colpa delle poste, (questo è da dimostrare - NdR), perchè continuo a ricevere senza ritardi altre riviste, si precisa che il n. 1 di 16/48 verrà duplicato sul n. 2 per tutti. Come dire: dopo il danno (del non caricamento di tutti i programmi), la beffa (di pagare due volte una cosa che altri che hanno saputo aspettare, pagheranno la metà).

Vi sembra corretto questo comportamento verso chi, come me ed altri, vi hanno dato fiducia ad occhi chiusi? Avete detto che so... quelli che hanno avuto difficoltà col n. 1 ce lo rispediscono, perchè è l'unica garanzia per sapere chi l'ha comprato realmente, e noi ve-

diamo di ritornarlo con l'aggiunta di qualche programma in più o addirittura del n. 2, come rimborso e scusa e avrete fatto il minimo del vostro dovere. E che l'errore non sia causa vostra non importa, perchè chi ne risponde al lettore siete voi.

E invece no. Non è colpa nostra, pagate e zitti, e se caso mai capita lo stesso col n. 2, raddoppiamo la dose. Così ci sarà gente che comprerà il n. 3 solo perchè è offerto come al supermercato: comprate 3 paghi 1, per non parlare di chi lo farà già dal n. 2 per lo stesso motivo.

Spero comprendiate che questo sistema, oltre che autolesionista, è al limite della truffa.

Io comunque, a questo punto aspetto. Aspetto di vedere il vostro comportamento in questa situazione. Inutile dire che se continuate su questa strada e non date soddisfazione a chi la deve avere per diritto, perderete un cliente nella mia persona e spero tanti altri che ragionano come me, e pensate alla bella pubblicità che faremo sulle vostre pubblicazioni.

Sperando su un risultato positivo a favore di chi vi segue fin dall'inizio, vi saluto cortesemente.

(M. Ballestri - Bologna)

Pubblichiamo integralmente, per non venire accusati anche di travisare il senso. Caro lettore, tra lasciamo i bytes più cafoncelli del suo text-file, e randomizziamo, astraeando un pb: in redazione tendiamo a riconoscere, a noi e gli altri, un diritto elementare, ci sembra anche umano, e cioè quello di sbagliare. Tant'è vero che, diabolicamente, pubblichiamo anche la sua lettera. E quanto al rime-

dio, era l'unico logisticamente praticabile.



Il caricatore di codice macchina

Sembra che il modo di caricare un programma in linguaggio macchina continui a essere un problema o un mistero per molti lettori. I sistemi sono due: caricare direttamente i codici (partendo dal formato decimale o esadecimale), o utilizzare un *Assembler*, un programma cioè che si sostituisce al basic, e consente di scrivere le istruzioni mnemoniche, provvedendo poi automaticamente alla compilazione in codice macchina.

Esistono molti *Assembler* per lo Spectrum, praticamente ogni grossa software-house ne ha uno: Hi-Soft, Picturesque, McGraw-Hill, Bug Byte, Softek, etc.; Ciascuno ha le sue specifiche d'uso, per cui è d'obbligo studiarli bene il manuale relativo.

Per caricare una routine già pronta si usano normalmente brevi programmi basic, noti come *hex-loader*, o semplicemente caricatori: *hex*, perchè nella maggioranza dei casi si utilizzano i codici in forma esadecimale, più pratica della decimale.

Ogni locazione di memoria di un byte contiene un numero lungo al massimo 8 bit (1 byte, appunto), il cui valore massimo è 1111 1111, che in hex è FF e in dec



255; il vantaggio pratico dell'hex sta qui: i codici si possono sempre rappresentare con 2 caratteri (notate che con numeri maggiori ne occorrono 3: 256d = 100h), facilitando la tabulazione e la lettura all'interno di una stringa. Per trascrivere il codice nella locazione di memoria abbiamo però a disposizione (in basic) solo POKE, il cui argomento dev'essere decimale: bisogna quindi trasformare l'hex in dec. Il programma che segue è già stato pubblicato più volte, a corredo degli articoli di G. Carri.

```

10 REM Caricatore di
   CODICI ESADECIMALI
20 CLEAR 59999: LET d=0: LET x
   =60000
30 LET a$=""... (qui settate,
   uno di seguito all'
   altro, i codici hex'
40 FOR a=1 TO LEN a$ STEP 2
50 LET b=CODE a$(a): LET c=CODE
   a$(a+1): LET d=b+c*d
60 LET b=b*48-39*(b/96): LET c
   =c-48-39*(c/96)
70 POKE x,b*16+c: LET x=x+1: N
EXT a
80 IF d<check THEN PRINT "er
   ror: di battitura in linea 30":
   STOP
90 SAVE "prog"CODE 60000,(LEN
   a$)/2

```

20: libera la RAM necessaria
 30: contiene una stringa con i doppietti hex (con le lettere minuscole)
 30: apre il ciclo di lettura, in funzione della lunghezza della stringa
 50: legge il codice ASCII di 2 caratteri della stringa; d è il check-control
 60: converte in decimale; i codici ASCII delle cifre sono da 48 a 59, le lettere minuscole da 97 in poi: ciò vi dovrebbe chiarire il senso delle sottrazioni
 70: scrive in locazioni di memoria consecutive i valori ottenuti e incrementa i contatori.
 80: verifica che il totale dato dal check-control sia esatto; bisogna ovviamente conoscere il check originale, solitamente fornito dall'autore del programma
 90: SAVE della routine allocata nella giusta destinazione.
 Le cose da non dimenticare mai sono il CLEAR prima di caricare il programma e un SAVE prima di lanciarlo: ciò eviterà di dover rifare tutto se qualcosa non funzionasse.

Logo per lo Spectrum

Vorrei imparare il linguaggio LOGO:

è reperibile la cassetta e/o il manuale? (M. Lattuada - Milano)

Per lo Spectrum esiste lo *Snail-Logo*, della CP, che il nostro severo censore Luigi Callegari ha valutato "povero" e ancora nel manuale.

Anchora e sempre Totocalcio

Su quali numeri è stato pubblicato il programma "Totocalcio"? (G. Ocellio - M. Marittima GR)

Sui nn. 4/5, e sul 6 il programma di aggiornamento. Ricordiamo che è disponibile anche su cassetta.

ZX81 in disarmo

Da qualche tempo non si trovano più accessori per lo ZX81. Pubblicherete lo schema per l'espansione di memoria? (I. Cassani - Milano)

Effettivamente sia gli accessori che il software per lo ZX81 stanno ormai scomparendo dai negozi. Il nostro consiglio è di sfruttare i piccoli annunci, gratuiti per tutti: molti possessori di Spectrum o di altri computer hanno ancora in fondo al cassetto uno ZX81 più o meno accessoriato. A questi chiediamo di non essere esosi nelle richieste: il valore di mercato di un materiale oggettivamente obsoleto non può essere quello originale, nemmeno deflazionato, e chiedere 200.000 lire per uno ZX81 16k, come appare in qualche annuncio, ci sembra decisamente scorretto.

Errore di caricamento

Salvato un programma su cassetta, il VERIFY dava "Tape loading error". Dopo aver provato più volte, ho resettato e caricato con LOAD "". Il programma viene caricato (?) in un tempo molto lungo, ma alla richiesta di LIST dà FOK senza mostrare niente... (S. Doria - Milano)

Ah, ah!... Resettare è stata un'azione decisamente incauta: se il VERIFY dà errore, si riavvolge il nastro e si ripete il SAVE! Anzi, si eseguono due SAVE! Che cosa carichi a questo punto il tuo Spectrum, non lo possiamo proprio sapere.

Numeri di linea e memoria libera

Come si ottengono righe di programma con lo stesso numero di linea? Perché PRINT 85535-USR 7962 (richiesta della memoria libera) dà, a computer resettato, 41472, e non 49152 (cioè 85535-16383)? (P. Miglioranza - S. Massimo VR)

Una linea di programma è composta da: 2 bytes per il numero di linea, 2 by-

tes per la lunghezza in bytes della linea (esclusi questi primi 4, compreso un ENTER finale), una quantità variabile di bytes con le istruzioni, un byte finale che è sempre 13 (ENTER). Appurato questo, si possono azzerare i numeri di linea delle prime righe di un programma con la seguente procedura:

```

100 LET inizio=PEEK 23635+256*P
   EEK 23636
110 FOR a=1 TO 10
120 LET lungh=PEEK (inizio+2)+
   56*PEEK (inizio+3)
130 POKE inizio,0: POKE inizio+
   1,0
140 LET inizio=inizio+lungh*4
150 NEXT a

```

È meglio usare il truccetto solo con linee REM: se il programma dovesse saltare a una delle linee alterate, il puntatore non la troverebbe più.

Quando alla memoria libera, 16384 sono i bytes occupati dalla ROM, 8912 il display-file, 768 l'area UDG, che sta sopra RAMTOP; la somma di questi è 24064, che sommato a 41472 danno 65536. Per avere i bytes realmente disponibili, ne andrebbero sottratti ancora 180 di variabili di sistema e una quantità dinamica usata dal calcolatore e i cui estremi sono E-LINE e RAMTOP. Infine, quello che resta è sì per il basic, ma compresa la variabile che il programma definisce.

Turbo-LOAD per ZX81

Esiste un programma di caricamento veloce per lo ZX81? (M. Zucchi - Modena)

Sappiamo solo di hardware che svolge questa funzione.

Ho tolto da un libro sul ling. macchina la seguente routine:

```

LD A,2
CALL 1601h
LD HL, 1821h
LD (23688), HL
LD A, (CODE carattere)
RST 10
RET

```

che dovrebbe scrivere il carattere, il cui codice è assegnato nella quinta istruzione, in AT 0,0. Ma non funziona. (M. Sorrentino - La Spezia)

A parte il fatto che non spieghi come l'hai caricata, e tralasciamo pure la nostra difficoltà a interpretare la tua grafia, il valore 10 associato a RST è esadecimale (quindi 16d).

Ecco come funziona: si carica in A il codice di flusso relativo al canale desiderato (in questo caso 2, per la parte alta dello schermo); la CALL esegue la

routine di OPEN, che apre il flusso verso il canale suddetto; si carica in HL la posizione di stampa - rileggetevi come funziona la var. S-POSN (23688/9); si trasferisce in queste locazioni il valore in HL; serve ora la routine di stampa PRINT (indirizzo 16), che essendo in pagina 0 può venire chiamata con un restart, risparmiando anche 2 bytes rispetto a CALL.

Inverse video XZ81

Ho applicato l'Inverse video allo XZ81. Tenendo «alta» la luminosità del tv, metà video resta più scura dell'altra. Perché? (W. Colla - Soluzzo CN)

All'oscuro siamo più noi del tuo video, non potendo esaminare la scheda (?) che hai usato.

Collegamento al monitor

Ho un monitor Hantarex con Ingresso RGB, e mi hanno detto che non è possibile collegarlo allo Spectrum (E. Orlandi - Pavia)

I monitor Hantarex solitamente sono dotati sia di Ingresso RGB (colori separati) sia PAL-CCIR (colori miscelati, equivalente al VCR del tv color), e lo Spectrum può essere collegato soltanto al secondo; se c'è solo l'RGB, niente da fare.



La memoria video

Perché il display-file è organizzato in modo così complicato nella memoria (A. Morreale - Ostia)

I progettisti avevano probabilmente qualche difficoltà da risolvere nell'hardware, quando hanno scelto di gestire il video in quel modo: sicuramente ne è risultata una gestione del software che non è delle più agevoli, ma il motivo non lo conosciamo.

Stampanti e interfacce

Che differenza c'è tra le stampanti con interfaccia Centronics e quelle con la RS232? E tra una stampante come la Seikosa e una da word-processing (in termini di software e di prestazioni)? (P. Ballacci - Monza MI)

La differenza non è nelle stampanti, ma nell'interfaccia: infatti la maggior parte delle macchine è disponibile, solitamente, dotata di porta tipo Centronics di serie e di RS232 opzionale. La Seikosa produce anche stampanti adatte al word-processing, e le differenze tra una

marca e l'altra, a parità di prezzo, non sono molto rilevanti, se il modello è generale purpose. Quanto al software, dipende anch'esso dall'interfaccia e non dalla stampante.

Come aumentare il BEEP dello Spectrum? È vero che i problemi di caricamento dipendono anche dalla polarità del registratore? (G. Celardo - Acerra NA)

a) Inviando il segnale, che può essere prelevato semplicemente dall'uscita MIC, a un amplificatore.

b) Sì: infatti recenti registratori, prodotti appositamente per gli home computer, sono dotati di un comando che inverte la polarità del segnale in uscita; ciò non sarà mai sufficiente, in ogni caso, a recuperare registrazioni troppo deboli e sporche.

Word-processing

Qual'è il word-processor usato per la pag. 22 di SC n. 10? Quale mi consiglia? Vorrei usarlo per la tesi di laurea: quale stampante usare? (M. Piacidi - Verona)

Non è disponibile, ma quasi tutto quello che fa si può fare anche con il Tawson II, di cui circolano anche versioni italianizzate.

Utilizzare un wp per stendere la tesi è un'ottima idea, che sembrerebbe logica e normale, ma informati bene, prima di lavorare per niente: sappiamo che in molte università, i vecchi gufi che le abitano non solo guardano storto, ma rifiutano addirittura le tesi scritte in dot matrix, e una printer a margherita non è alla portata di tutti. Non sappiamo se il problema sia solo estetico (sic!), o se persista qualche ancestrale paura di queste macchine diaboliche...

È preferibile comprare un'interfaccia joystick tipo AGF o Kempston, oppure una programmabile? (Danilo Neri - Roma)

Quelle programmabili sono senza dubbio più versatili.

Ho uno Spectrum espanso a 80k; vorrei sapere quanti kbytes ho a disposizione, come si collega la Epson RX80, se la Superface Sandy è compatibile con il MicroSpeech Currah. (Andrea Mazzarello - Genova S.)

hai a disposizione gli stessi kb di un 48k, più un «banco» di 32k, inutilizzabile dal basic, in cui teoricamente si possono mettere dati o c.m., paginando la RAM. In pratica l'operazione era semplice solo con la ISSUE2.

La stampante si collega come tutte le altre stampanti (quante volte dovremo ripeterlo ancora?), cioè con un'interfaccia parallela tipo Centronics o con la porta seriale dell'Interface 1.

Le compatibilità di espansioni diverse dipendono dall'uso di EPROM o dal software che richiedono: non si devono, ovviamente, sovrapporre nelle stesse aree di memoria. Consultare le istruzioni e provare: al massimo non succede niente.



Esperimenti

Con il seguente programma

LD A, 7Fh

RST 10h

RET

Si ha un messaggio di errore, mentre con

LD A, 2h

CALL 5633d

LD A, 7Fh

RST 10h

RET

tutto va alla perfezione. Perché? (Giuseppe Reale - Milano)

Quale messaggio di errore? A noi risulta che dia un OK senza stampare niente, poiché non viene aperto il canale video (confronta la risposta data a un quesito in qs. stesso numero), il secondo programma apre questo canale, e stampa il CHR\$ 127 (7Fh), cioè il Copy-right.

Bootstrap

A cosa servono le EPROM? È possibile far sì che appena acceso lo Spectrum esegua il load di qualche programma? (G. Tassone - Reggio c.)

Sulle EPROM stiamo pubblicando, come avrai visto, una trattazione esauriente. Il «bootstrap» automatico non è possibile con lo Spectrum; con i microdrives, si può avere un programma sul mdv1 (solo su questo) salvato col nome «run», che viene caricato dando semplicemente il comando RUN, all'accensione del computer.

Altri esperimenti

Provate questo:

10 LET C\$ = INKEY\$

20 IF C\$ = "" THEN GO TO 10

30 PRINT C\$

40 GO TO 10

Date RUN, quindi inserite il CAPS LOCK; ora premete ENTER. Perché ap-



RESISTENTI, COMPATTE SILENZIOSE, EFFICIENTI, AFFIDABILI E COMPETITIVE.

Stampanti MITSUI con le caratteristiche tipiche dei giapponesi.



La TELCOM propone una gamma di stampanti che si distingue per la varietà delle funzioni e per la grande affidabilità:

- 80 e 132 colonne
- 120 e 180 caratteri al secondo bidirezionale ottimizzata
- fogli singoli, moduli continui
- vari tipi di caratteri
- stampa espansa, compressa, NLQ, grafica
- interfaccia parallela, seriale, buffer fino a 128 KB, interfacce speciali.



gioca la carta
telcom



paiono quelle scritte? (Giuseppe Balderi - Roma)

Come si introducono le linee 0? Provatelo questo:

```
10 POKE 23613,0
20 PRINT "(qualsiasi)"
30 GO TO 10
```

DATE RUN, quando appare «scroll?», premete contemporaneamente i due shift, comparirà ancora RUN, con il cursore (ovviamente) in «E»; premete ENTER e...? (Francesco Trevisanni - Cerignola FG)

Ripetiamo ancora una volta la storia della linea 0: senza Interface 1, si da POKE 23755,0: POKE 23756,0

Con l'Interface 1:
LET q = PEEK 23635 + 256 * PEEK 23636

POKE q,0: POKE q + 1,0

Gli strani risultati (uguali tra loro), a cui si riferiscono i due lettori, sono dovuti a uno dei tanti bug di sistema, in cui si entra con digitazioni sintatticamente non previste dal sistema operativo. Questo in particolare, non serve a niente (se trovate un uso scriveteceelo), ed è innocuo, essendo impossibile entrarci involontariamente. (Ma se ci entrate, spesso avviene un reset distruttivo).



Caricamenti multipli

Vorrei chiarimenti sul modo di caricare informazioni dalla cassetta (blocchi Bytes) senza rovinare un programma già nella memoria dello Spectrum (C. Cavalli - Milano)

Sinteticamente:

a) progr. basic in memoria, e cod. macchina da caricare:

```
CLEAR x: LOAD "nome" CODE x + 1
```

b) progr. in c.m. in memoria e altro c.m. da caricare; posto che il primo inizi a x e sia lungo y, e il secondo sia lungo z, si hanno due casi:

b1) il secondo va messo prima:

```
CLEAR x-z: 1: LOAD "nome" CODE x-z
```

b2) il secondo va dopo:

```
LOAD "nome" CODE x + y + 1
```

Tutto ciò dopo aver appurato la compatibilità del due programmi ed eventualmente la loro riocciabilità.

Giochi e Gioystick

Si può usare un'interfaccia joystick diversa da quella prevista, modificando il software? Che valore ha il byte che il processore legge sul bus dati se si ha

un'interfaccia Kempston inserita? (C. Padronetti - Terni)

Molti videogames sono sprovvisti dell'opzione joystick: come si possono adattare? (F. Rossi - Genova)

Come utilizzare un joystick in alternativa alla tastiera? (R. Macina - Treviglio BG)

Se un videogame non prevede l'opzione joystick o la prevede per un certo tipo di interfaccia, intervenire sul programma per poter usare qualcos'altro è un lavoro da super esperti: dovete "soltanto": a) sprotteggere il programma, in genere tutto in ling. macchina e con autostart; b) passarlo con un monitor di assemblatore fino a individuare dove legge i valori che l'interfaccia invia sul bus dati; c) modificare gli indirizzi da leggere in funzione dell'interfaccia usata (che dovete ovviamente conoscere); se vi sembra facile...

Per usare un joystick in alternativa alla tastiera, in ambiente basic, è abbastanza facile: anche qui occorre conoscere gli indirizzi usati dall'interfaccia sul bus dati. Cfr. in questo stesso numero la rubrica «Un tasto per volta», a proposito della funzione IN.

SAVE di matrici

Come si registra un archivio su nastro? Se a un array registrato voglio aggiungere dei dati, lo Spectrum cancella quelli che ho, oppure scrive i nuovi dopo i vecchi e non registra questi ultimi... posso risolvere il problema con i Microdrives? (Marco Giorgi Piombino LI)

Come si salvano le matrici di dati, e come si leggono una volta ricaricate? (C. Pacetti - Bolzano)

Per salvare una matrice esiste un comando apposito:

```
SAVE "nome" DATA a()
```

Il DATA è il comando ottenuto con il tasto "D" in modo esteso; vale anche per le matrici di stringhe, e non serve specificare le dimensioni. Per ricaricare, basta eseguire

```
LOAD "nome" a()
```

anche senza aver in precedenza initializzato la matrice con DIM; è pure possibile ricaricare l'array con un nome diverso da quello originale.

L'inconveniente di questo sistema è che le dimensioni sono fisse: una volta che la matrice (con qualsiasi numero di dimensioni) sia stata riempita, non si possono aggiungere elementi. Il trucco per aggirare il problema però esiste: fate una copia di sicurezza del programma, senza dare CLEAR (registrate anche le variabili); da ora in poi non editate più linee di programma, né fatele girare fino al termine delle operazioni. Calcolate l'ingombro dell'array: se è di

caratteri, come q\$(x,y), fate il prodotto x*y, se è numerico; q(x,y), fate x*y*5; ottenete un risultato yy.

Scrivete questa linea di programma, con un qualsiasi numero di linea che non abbiate usato:

```
9999 LET q(1,1) = q(1,1): PRINT PEEK 23629 + 256 * PEEK 23630: STOP
e date GO TO 9999; q (o q$) dev'essere il primo elemento del vostro array. Sullo schermo viene stampato un numero xxx. Digitate
```

```
SAVE «matrice» CODE xxx,yy
```

fate anche un VERIFY, per maggior sicurezza, e ridimensionate l'array, ingrandendolo della quantità desiderata, con

```
DIM q(x,y,1)
```

DATE nuovamente GO TO 9999: otterrete un valore zzz; per finire, ricaricate i vecchi dati con

```
LOAD «matrice» CODE zzz
```

Verificate che l'array contenga i dati e lo spazio per i nuovi inserimenti.

La stessa procedura si può seguire con i microdrives, con i quali però si può fare molto di più, come vi raccontano due nostri collaboratori poche pagine più avanti.



(F. Bertani - Milano) L'immagine spostata a sinistra è normale nell' Issue 3, sulla quale non si può intervenire per regolare il colore: si può agire solo sul tv. La GP50S con il word pr. stampa solo a 64 colonne.

(G. Vassallo - Torino) L'ingresso per videoregistratore del tv color ha una linea sola che porta il segnale "composito" (se l'ingresso fosse RGB, non si può collegare), e una GND (massa); trova i piedini giusti sullo schema del tv, noi da qui non possiamo vederli.

(M. Santini - Sesto S.G. MI) I word processor sono previsti per essere usati con una stampante "grande", la stampa su Alphascom o GP50S è solo accessorio. Cambiare il numero di car. per riga non è possibile.

(L. Raffa - Verbania NO) La distribuzione del software in Italia continua ad avere una gestione deprecabile: chi si dichiara "ufficiale" ha un aggiornamento spesso ridicolo, per cui, piaccia o no, per avere rapidamente un programma, specie se nuovo, restano i canali alternativi...



L. Cristalli - Piacenza) Più no che si, in realtà fa quasi tutto il computer. Non vendiamo cassette, escluse quelle prodotte da noi.

(R. Ricciardi - Napoli) Se rileggesti con più attenzione quella pagina, ti accorgeresti che è un'inserzione pubblicitaria: il concorso era gestito dalla società Inglese a cui andavano inviate le partecipazioni. Noi non c'entriamo per niente.

(E. Giorelli - Catania) Gli indirizzi forse li pubblicheremo, un giorno. Entrambe le case citate producono software per Spectrum.

(M. Catzone - Milano) 1. Sì, ma può essere residente nell'interfaccia; 2. Sì, dotando la stampante di porta seriale.

(C. Garofalo - T. del Greco NA) Ti consigliamo di consultare i piccoli annunci alla fine della rivista, contattando inserzionisti della tua zona.

(M. Asquini - Roma) 1. No; 2. Sì: il van-

taglio c'è se possiedi già quella espansione, altrimenti è solo per chi la vende.

(R. Magro - Cordenons PN) Per il READ/DATE su ZX81 la versione corretta è sul n. 11, ci scusiamo di nuovo per l'errore ripetuto. La ROM è stata pubblicata in ordine di byte, non si possono seguire i salti in avanti di una routine.

(V. Riguzzi - Bologna) Purtroppo temiamo di sì: all'inizio di una registrazione si trova un header, senza il quale il computer non può identificare le caratteristiche del programma che aveva salvato.

(F. Schinco - Torino) Non possiamo accontentarti, anche perché non esiste UN metodo, ma declino. Diversamente, come?

(M. Poletto - Bolzano) Sicuro lo è, basta che non si cerchi di fargli fare cose per cui non è stato pensato. La seconda notizia è destituita di qualsiasi fondamento.

(P. Ciavattella - Pescara) È probabile

che la colpa sia dello spinotto di alimentazione; ma come puoi lavorare senza registratore? I listati vanno dattiloscritti, se non hai la stampante; ma che conta è la cassetta. Salvando nuovamente il programma senza dare CLEAR. Manca lo SCROLL perché il programma è scritto per lo Spectrum. La ZX-printer è fuori commercio; in ogni caso, meglio l'Alphacom.

(A. De Gregoris - Milano) 1. Sì, ma è più prudente una protezione dalle sovra-tensioni. 2. Domanda incomprensibile: non serve interfaccia, servirebbe più RAM. 3. Se il videogame prevede il secondo giocatore e se l'interf. joystick è di tipo previsto. 4. Rai Tre, al pomeriggio.

(M. Ruggeri - Pesaro) 1. Sono due caratteri grafici, non fondamentali. 2. e 3. Le modifiche apportate non sono pertinenti né corrette: copia il programma così com'è e vedrai che funziona; in particolare, hai cambiato arbitrariamente le linee 8 (manca RESTORE), 20 (manca uno SPACE alla fine), 25 (manca una PRINT), 32, 40, 82 (manca un salto). Scusa il ritardo.

dall'INGHILTERRA i fantastici computer games
MASTERTRONIC

**ELETTRIZZANTI
AVVINCENTI
EMOZIONANTI**

**4 NOVITÀ
OGNI MESE**
dal tuo
rivenditore
di fiducia.



CEM64



SPECTRUM
CEM64



CEM64



SPECTRUM
MSX

**TUFFATI NEL FANTASTICO
MONDO MASTERTRONIC!**
per vivere nuove emozionanti
avventure piene di suspense e
frenetiche animazioni.

QUALITÀ - PREZZO **solo £ 7900** è la grande proposta
MASTERTRONIC per conquistare tanti amici.

Mastertronic s.a.s. - V.le Aguggieri, 62/A - 21100 Varese - ☎ 0332-236698

Provando e riprovando, due collaboratori hanno scoperto come fare cose turche con i microdrives

Accesso casuale ai Microdrives



a cura di R. Zonin e D. Bolla

I Microdrives rappresentano un notevole miglioramento rispetto alle normali cassette audio, sia in termini di velocità che di affidabilità. Purtroppo, essendo anch'essi dispositivi a nastro, mantengono uno dei principali difetti delle cassette: l'impossibilità di accedere in modo RANDOM (casuale) ai dati contenuti. Vedremo più avanti che un modo per implementare l'accesso RANDOM esiste, prima però, consentiteci qualche cenno teorico.

Accesso sequenziale e accesso Random.

In un dispositivo ad accesso sequenziale non è possibile accedere direttamente a un qualsiasi dato memorizzato in un file, in quanto si dovranno prima leggere tutti i dati che lo precedono. L'accesso, quindi, avverrà sempre leggendo il primo record, poi il secondo, e così via fino al record che ci interessa.

È evidente che questo modo di cercare i dati è decisamente lento. Un tipico dispositivo ad accesso sequenziale è il nastro in cassetta. L'accesso sequenziale è, in un certo senso, fisicamente connesso al nastro.

Nei dispositivi ad accesso random o casuale è possibile andare a leggere direttamente il dato che ci interessa, saltando tutti quelli che lo precedono. Questo consente una ricerca più rapida e flessibile, anche se la gestione diventa un po' più complessa.

I più diffusi dispositivi ad accesso random sono i floppy disk. I drive per dischi sono piuttosto costosi, e difficilmente alla portata dell'utente medio dello Spectrum.

Ai Sinclairisti che vogliono provare le tecniche di accesso random sul proprio sistema, proponiamo un metodo per trattare in questo modo il loro microdrive.

Il Microdrive

Il supporto sul quale il microdrive registra i dati è una cartuccia di nastro che presenta due differenze sostanziali rispetto alle cassette audio. La prima è che il nastro è costituito da un anello chiuso (v. S.C. n. 6 p. 53), che continua a girare davanti alla testina di lettura, come nelle vecchie cassette stereo - 8 (qualcuno le ricorda?).

La seconda fondamentale differenza è che la cartuccia del micro-



drive viene formattata. Ciò significa che, prima di poter contenere dei dati, la cartuccia è stata divisa in tante zone, dette settori.

In ogni settore vengono memorizzati un numero progressivo, che lo identifica univocamente, e una serie di informazioni di controllo. Ognuno di questi settori è destinato a contenere un record di dati, il quale record a sua volta sarà numerato per renderlo riconoscibile univocamente all'interno di un file.

Il trucco c'è

A questo punto dovrebbe essere chiaro che, un volta nota la posizione di un dato all'interno di un file, niente ci impedisce di calcolare in che record il dato stesso risieda. E una volta scoperta la posizione logica del dato all'interno del file (cioè il numero del record) lo Spectrum è in grado di trovare la posizione fisica nella cartuccia (cioè il numero del settore). Tutto quello che dobbiamo fare per accedere casualmente ai nostri dati è "far credere" al sistema di avere già letto tutti i dati precedenti. Come tutti gli "smanettoni" avranno già capito, stiamo per andare a curiosare nelle variabili di sistema.

Dove mettere le mani

Le variabili che ci interessano non sono le solite variabili di sistema, ma informazioni che si creano solo quando deve essere usato il microdrive. Esse si trovano in un'area chiamata *Canale dei microdrive*, che generalmente inizia a partire dall'indirizzo 23844, all'interno dell'area chiamata *Informazioni di canale*, puntata dalla variabile CHANS.

Per quel che riguarda la lettura dei dati, sono 2 le variabili che ci scrivono:

CHREC (locazione 23857): indica il numero progressivo del record appena letto all'interno del file, nel range 0 + 255 (un file è lungo quindi 128 Kb al massimo; almeno in teoria, perché in pratica le cartucce non contengono più di 100 Kbytes).

CHBYTE (locazioni 23855/23856): punta al prossimo carattere da leggere del record: la lettura dal micro-

drive avviene infatti per blocchi di 512 bytes, che vengono memorizzati in un buffer di RAM. Questa variabile indica quindi a che punto siamo con la lettura all'interno del record.

Che cosa fare

È chiaro adesso che per trovare un certo dato basta conoscere la sua posizione all'interno del file (Intesa come distanza dal primo carattere). Già, direte voi, ma la posizione come la troviamo?

Ci sono diversi modi, a seconda di come sono memorizzati i dati. Supponiamo per esempio di avere memorizzato dati di lunghezza variabile.

In questo caso l'unica possibilità è di riprodurre in un array nella RAM il punto di inizio di ciascun dato, ed utilizzare l'array per la ricerca. Il listato 1 mostra un semplice esempio di questo metodo.

Naturalmente questa non è la soluzione ottimale, soprattutto perché obbliga a tenere in memoria una tabella di informazioni ingombrante e tutto sommato inutile, in quanto non contiene dati per noi significativi (in un archivio indirizzi sono significativi "cognome" e "telefono",

ma non "posizione occupata nel file").

L'unico modo per eliminare questo array è rendere standard la lunghezza di ogni dato. Mantenendo fissa questa lunghezza all'interno del file, diventa facile calcolare la posizione delle informazioni che ci interessano. Per esempio, in un archivio indirizzi dove ogni dato è lungo 80 caratteri, il 12esimo indirizzo comincerà al $(80 + 12 =)$ 960esimo carattere. Potete vedere un esempio di questo metodo nel listato 2.

Già, obletterete, è semplice trovare un dato conoscendo il suo numero d'ordine. Ma questo numero come lo trovo? Non posso cercare il telefono del signor 15 o la via della signorina 61.

Giustissimo. Abbiamo dunque bisogno di qualcosa che legni il dato che stiamo cercando al suo numero d'ordine, che a sua volta individua la sua posizione nel file. Abbiamo bisogno della cosiddetta *chiave*.

La chiave

Prima di vedere cosa si intende per chiave, esaminiamo più in dettaglio il metodo di ricerca su file che vogliamo realizzare.

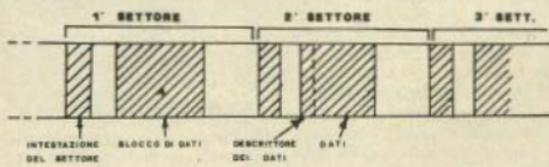


FIG. 1

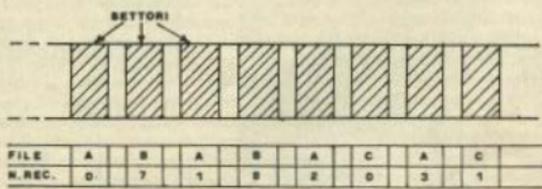


FIG. 2

Supponiamo di avere il nostro solito archivio indirizzi, e di voler stampare i dati riguardanti il nostro amico Pinco Pallino.

Alla richiesta del computer "record da stampare?", risponderemo "Pinco Pallino". A questo punto il computer dovrà cercarsi il numero corrispondente a Pinco Pallino, far partire il microdrive, leggere i dati memorizzati e stamparli.

Il campo che la macchina ha usato per la ricerca è chiamato *campo chiave* (o semplicemente *chiave*), e il suo contenuto fornisce una chiave di accesso a un gruppo di dati (per esempio, a un indirizzo). Notate che la chiave di accesso al dato è una parte del dato stesso.

In particolare, sarà la parte che ci è nota. Così in un indirizzario la chiave potrà essere il cognome e nome, in uno schedario dischi sarà il titolo del long playing; è importante comunque che la chiave individui univocamente il dato a cui si riferisce.

La parte del dato scelta come chiave verrà caricata in memoria centrale prima della ricerca, che verrà effettuata quindi in RAM e molto più velocemente che su microdrive; una volta trovata la posizione del dato richiesto, si carica dal drive la parte mancante. Riprendendo l'esempio precedente, il programma caricherà in RAM tutti i *nome & cognome* contenuti nei file, mettendoli in un array; quindi esaminerà l'array, alla ricerca del nome che gli abbiamo dato da trovare; localizzato quest'ultimo nell'array, si moltiplicherà la sua posizione per il numero di caratteri occupato da ciascun indirizzo ed avremo la posizione dell'indirizzo completo nel file.

Con questa informazione, il programma caricherà dal microdrive il resto del record: via, città, telefono ecc. Il listato 3 mostra un esempio di questo modo di operare.

Un'ultima osservazione: con questo metodo possiamo eseguire ricerche su un file soltanto attraverso il campo chiave che abbiamo scelto; usando come chiave *nome & cognome* non possiamo fare ricerche per città o via.

In seguito vedremo come supera-

Glossario

RECORD: dobbiamo distinguere il record *fisico* ed il record *logico*. Per record fisico intendiamo un blocco di 512 bytes di dati, più una serie di informazioni di controllo, che viene trasferito da o verso microdrive in una singola operazione di lettura o scrittura. È sinonimo di blocco dati.

Per record logico invece intendiamo un gruppo di dati che si riferiscono allo stesso soggetto. Per esempio, in un'agenda telefonica, il record logico di ogni persona conterrà *nome, cognome, prefisso, numero telefonico*: in un archivio libri conterrà *titolo, autore, anno, casa editrice*.

CAMPO: per campo di un record si intende uno dei dati che il record potrà contenere; ad esempio, il campo *città*, il campo *titolo*, ecc. Spesso si indica con *campo* anche lo spazio fisico dove si può scrivere un dato. Se per esempio abbiamo deciso che il campo *città* sia di 10 caratteri, e inseriamo "Milano", verrà registrato comunque un gruppo di 10 carat-

teri, anche se il dato ne occupa solo 6.

DATO: ha un significato piuttosto vasto; noi lo useremo per indicare il contenuto di un record logico o di un campo di record. Quindi un dato potrà essere l'indirizzo di una persona, il titolo di un libro, ecc.

FILE: per file intendiamo una collezione di record. La macchina lo vedrà come una serie di record fisici (*blocchi*); noi invece come una serie di record logici (*dati*).

BLOCCO DATI: il blocco minimo che il nostro Spectrum può leggere o scrivere ed è pari a 512 bytes. Notate che, mentre noi possiamo creare record grandi quanto vogliamo, lo Spectrum carica e salva solo 512 bytes per volta.

FORMATTAZIONE: un'operazione durante la quale lo Spectrum ricerca i blocchi difettosi all'interno della cartuccia e li segna in modo da non usarli. La capacità teorica possibile di una cartuccia è di 130560 bytes, ma purtroppo in media ne vengono utilizzati circa 93000.

re questa limitazione con l'uso delle *chiavi multiple*.

Commenti

Come già detto, esistono due modi di registrare i dati: uno che utilizza la matrice per sapere dove si trova il dato e un altro che utilizza record di lunghezza fissa e che quindi non ha bisogno della suddetta matrice per ricercare i dati al suo interno.

Il metodo a matrice (listato n. 1) ha il vantaggio che i dati occupano solo lo spazio necessario a contenerli e non un byte di più; inoltre i dati possono assumere una qualsivoglia lunghezza senza avere problemi. Lo svantaggio è che si è costretti ad avere in memoria una matrice di dimensioni notevoli; inoltre

la modifica di files con questo tipo di sistema risulta agevole solo se si hanno a disposizione due microdrive.

Il metodo a lunghezza fissa (listato n. 2) è standard, non è eccessivamente difficile da implementare, non occupa memoria con matrici strane, può essere usato con un solo microdrive.

Il lato negativo è che la lunghezza dei dati deve essere definita e gli eventuali spazi inutilizzati vengono scritti sul microdrive come gli altri utilizzati; questo provoca uno spreco di memoria all'interno della cartuccia.

Come lo Spectrum gestisce i dati

Qui per dati non intendiamo solo



I dati relativi a un database, ma qualsiasi genere di informazione che si voglia memorizzare (programmi, videate, c. macchina, etc.).

Guardiamo la figura 1. Un settore è costituito da un blocco di intestazione che contiene il numero di settore e il nome della cartuccia; c'è poi una zona libera e infine abbiamo il blocco dati, che contiene un *descrittore* di record, nel quale c'è scritto il numero del record, la sua lunghezza, il nome del file al quale si riferiscono quei dati; c'è anche un byte di controllo che serve al computer per sapere se i dati che ha caricato sono giusti.

Seguono poi i dati che a noi interessano. In figura 2 si può osservare come lo Spectrum, quando memorizza i programmi, non utilizzi mai settori contigui: generalmente scrive un settore sì e uno no. Ciò fa sì che i dati vengano distribuiti più uniformemente all'interno del nastro.

Grazie a questo accorgimento il tempo medio di accesso è pari a circa 4 secondi e il tempo di caricamento per blocchi non molto lunghi è pressoché istantaneo; questo rende i microdrive meno lenti di quanto qualcuno voglia far credere.

Nel maneggiare le cartucce, dovete avere una certa cura: non si deve mai staccare la corrente con una cartuccia inserita e soprattutto non si deve mai *attaccare o staccare* una qualsiasi periferica quando il computer è acceso, pena la morte di qualche elemento del sistema.

Questo è quanto in realtà si deve fare con qualsiasi altra macchina: un floppy-disk è forse più delicato di una cartidge. Se qualche problema può sorgere (a causa dei produttori di cartridges) è che, una volta formattata, la cartuccia dia valori di capacità estremamente bassi (minore di 48K).

Anche a questo si può rimediare, semplicemente formattando più volte, fino a 15-20, la cartuccia stessa: vedrete che i "Kappa" verranno recuperati. Non abbiate timore di continuare a formattare, finché non raggiungete un risultato soddisfacente (verificabile con CAT).

Per utilizzare i programmi

(list a pag. 38)

Informazioni generali

Questi programmi contengono la struttura base della ricerca random e quindi possono essere utilizzati da coloro che hanno un minimo di dimestichezza con l'interfaccia 1.

Ricordatevi che i files aperti dai tre programmi hanno lo stesso nome (che può essere cambiato a piacere, a patto che ve lo ricordiate quanto effettuate la ricerca); inoltre i tre files generati sono generalmente *incompatibili* fra loro.

Inoltre, visto che per ora si utilizza un programma semplificato, abbiate cura che il canale del dato aperto sia solo uno, e solo quello che vi interessa; in seguito vedrete che si possono gestire più files contemporaneamente.

Non preoccupatevi se per ora non si può modificare il file creato, ci abbiamo già pensato e lo vedrete in un prossimo numero; per ora cercate di capire come funzionano i tre listati.

Listato n. 1

Per far partire il listato 1, dovete inserire una cartuccia formattata nel microdrive n. 1 e dare GOTO 35.

Il programma vi chiede di inserire dei dati... inseritene in numero tale che il microdrive parta almeno una volta.

Non preoccupatevi se dovete digitare parecchio infatti questa versione salva soltanto quello che digitate e poiché è il buffer di canale è di 512 bytes...

Quanto avete finito date GOTO 1000 e il computer vi chiederà il numero di record. Digitatelo (non date numeri strani) e il computer vi darà il contenuto di quel record;

Per potere rivedere i dati dopo averli inseriti e dopo avere spento il computer, dovete salvare il programma *senza* dare CLEAR, e farlo ripartire con GOTO 1000 (per salvare la matrice).



Listato n. 2

Il listato 2 si lancia con RUN.

I dati inseriti non devono superare i 20 caratteri, a differenza del listato precedente nel quale ogni campo poteva avere lunghezza qualsiasi.

Quando avete finito, date un RUN 1000, notate che non ci sono più variabili strane da salvare, a questo punto vi sarà richiesto il n. del record che volete vedere e poi vi sarà mostrato.

Listato n. 3

Il listato 3, oltre a essere più completo, ha la lunghezza del campo non più limitata a 20 caratteri; inoltre, è possibile modificare i parametri.

Si parte con RUN e si scelgono le varie opzioni. L'opzione 3 utilizzata quando avete ben capito a che cosa servono le variabili.

Lanciare prima l'opzione 1: se il computer non trova un altro file con lo stesso nome, siete pronti per inserire i dati. Notate che se scrivete più caratteri di quelli consentiti questi saranno brutalmente tagliati.

Quando avete finito, scegliete l'opzione 2, digitate il COGNOME del record che cercate, e vi verrà dato il record completo.



È un disassemblatore pratico e affidabile, completamente in linguaggio basic, ma ciò nonostante, contenuto e compatto, completo e ragionevolmente veloce.

Un disassemblatore è uno strumento molto utile per decodificare routines in codice macchina residenti in ROM/RAM, e anche per verificare il corretto caricamento in memoria di programmi in linguaggio macchina: non utilizzando un programma assembler è facile commettere errori di codifica da istruzione simbolica a istruzione in codice, o errori di digitazione nell'immissione del codice macchina con comandi DATA; un altro errore molto banale si verifica nel calcolo di indirizzi, assoluti o relativi, nelle CALL e nelle istruzioni di salto.

L'output prodotto dal programma consiste in:

— indirizzo (decimale) dell'istruzione

— istruzione in codici macchina (decimale o esadecimale, secondo una scelta iniziale)

— istruzione simbolica

Gli indirizzi contenuti in istruzioni JP e CALL vengono espressi in decimale.

Gli indirizzi relativi contenuti in istruzioni JR e DJNZ vengono espressi col loro valore relativo, positivo (+) o negativo (-), e col corrispondente indirizzo assoluto.

Le istruzioni RST 8 e RST 40 vengono identificate, e i bytes seguenti, a loro attinenti, vengono espressi come DEFB (*define byte*)

Anche eventuali locazioni di memoria non contenenti un codice operativo valido, sono trattate come DEFB e, oltre il valore numerico, viene espresso anche il corrispondente carattere ASCII (o *token*, se CODE maggiore di 32).

Nella fase iniziale viene richiesto di immettere gli indirizzi di inizio e fine dell'area di memoria che si vuole disassemblare; quindi, è la data la facoltà di scegliere il tipo di output:

H = esadecimale

ENTER = decimale

B = bytes: ciascuna locazione dell'area richiesta viene disassemblata come DEFB; ciò permette di ottenere listati più intellegibili di

Un completo disassemblatore per lo Spectrum

di Giovanni Mellina

aree non contenenti istruzioni (costanti, messaggi, etc.).

Durante l'esecuzione, la pressione del tasto "R" comporta un *break*, con restart automatico dalla fase iniziale.

L'output è normalmente inviato su schermo; desiderandolo su stampante, è sufficiente, prima di eseguirlo, dare un comando OPEN 2, "p".

A fine esecuzione ricordarsi di ripristinare il corretto canale di output con

CLOSE 2.

Chi utilizza stampanti a 80 colonne, avrà qualche problema con la tabulazione, poiché il TAB non viene riconosciuto.

Il programma può essere salvato con l'opzione LINE 1: a questa linea corrisponde un RUN 1000, dove viene eseguito il caricamento iniziale delle tabelle e delle altre costanti utilizzate.

Dopo un BREAK o uno STOP, il programma può ripartire correttamente con GO TO 10 (non usare RUN 10); RUN o RUN 1 eseguono una partenza corretta, ma con completa riesecuzione del caricamento delle tabelle.

Queste tabelle sono due:

— t\$ per la traduzione delle istruzioni

— o\$ per la decodifica dei codici operativi,

caricate con DATA in fase di inicializzazione alle linee 1000 e seguenti.

Così strutturato, il programma non è eseguibile su CPU 16k, lasciando disponibile la memoria a partire dall'indirizzo 34100 (vedi CLEAR 34099 in linea 1000).

È possibile ottenere una versione ridotta, caricando separatamente almeno la tabella istruzioni t\$, ed eliminando tutti i REM: digitare a parte i DATA relativi (linee 1020+3022), salvare su cassetta la tabella, e inserire nel programma un LOAD""DATA della stessa.

L'affidabilità del disassemblatore è garantita al 100%, essendo stato sottoposto a test globale, disassemblando l'intero set di istruzioni Z80: 696 istruzioni, pari a 1416 bytes di cod. macchina.

Può verificarsi la necessità di disassemblare una routine o un programma, il cui indirizzo di caricamento sia inferiore a 34100: alla linea 1001 è prevista una variabile (LET d=0) che, opportunamente impostata, consente di aggirare l'ostacolo.

Per esempio: supponendo di dover disassemblare un programma non rilocabile, previsto per essere caricato dall'indirizzo 28500, dove: caricarlo in memoria con LOAD "nome" CODE 34500 e impostare un *displacement* correttivo pari a 34500-28500 (LET d= -6000); disassemblate ora le locazioni 34500 e seguenti, contenenti il programma: otterrete un output con gli indirizzi correttamente aggiustati.





STUDIO D
PER NON SMARRIRE MAI IL FILO DEL DISCORSO.
STUDIO D
ATTENTI RADIOTELEVISIVE INDIPENDENTI CHE SI FANNO SENTIRE.



CONCESSIONARI MEZZI
RADIOTELEVISIVI

STUDIO D
Via Rossini 5 - 20122 MILANO
Tel. (02) 799.592-782.503



Sul retro del nostro computer, si trova un connettore a 28+28 vie, dove sono disponibili i segnali per le periferiche e le espansioni del sistema.

Il pettine è ricavato direttamente dal circuito stampato del computer ed è diviso in parte alta (A), o lato componenti, e bassa (B), o lato software.

Oltre ai dati ed agli indirizzi, sono disponibili i segnali di controllo provenienti dalla CPU e dall'ULA, più un piedino per disabilitare la ROM, le alimentazioni e alcuni segnali per il video e il reset del sistema.

Cerchiamo di esaminare i vari segnali; sarà opportuno un cenno anche sui dati e sugli indirizzi, anche se probabilmente alcuni lettori li conoscono già (non siate impazienti, non tutti sono esperti in microprocessori).

Vi sono otto terminali chiamati D0 + D7, che costituiscono il *bus dei dati*; sono bidirezionali *three-state*, cioè possono servire da *input* per i dati o le istruzioni e da *output* dopo l'elaborazione dei dati.

Il bus dei dati della CPU può, andando in *three-state* (o alta impedenza), isolarsi dal mondo esterno; questa operazione è necessaria, per esempio, tra due operazioni consecutive, per consentire agli indirizzi di stabilizzarsi e non creare conflitto di dati tra un'operazione e l'altra.

Otto bit di dati costituiscono un byte, ogni byte può contenere un numero compreso tra 0 e 255 (2 all'ottava potenza); la CPU vede questi numeri come dati veri e propri o come istruzioni, a seconda dell'ordine in cui gli si presentano. Dopo alcuni cicli macchina, ripresenta sul bus il risultato dell'elaborazione per un successivo utilizzo.

I terminali A0 + A15 sono uscite *three-state* attive alte. Costituiscono il bus degli indirizzi, e la combinazione di «0» e di «1» su queste uscite seleziona quale locazione di memoria o quale dispositivo di I/O è interessato allo svolgimento del programma in quel momento.

Le combinazioni possibili sono 65536 con accesso alla memoria, o 256 con accesso agli ingressi/uscite.

EPROM/II

Teoria e pratica della EPROM

di Sergio Coraglia

Il connettore posteriore e i segnali disponibili

INT, NMI e BUSRQ significano rispettivamente: *Interrupt, Not Maskable Interrupt e Bus Request*; sono richieste di interruzione alla CPU attive a «0» (basse), come indica la sopraincisa.

A queste linee possono essere collegati diversi dispositivi hardware di input/output; quando uno di questi necessita di un servizio in breve tempo, invia un segnale a una di queste linee.

Il BUSRQ ha la priorità maggiore: attivando questa linea, tutti i bus della CPU vengono mandati in *three-state*, ed è possibile lo scam-

bio di dati tra la memoria e il dispositivo che ha richiesto il BUSRQ, in questo caso la CPU, che interrompe immediatamente l'operazione in corso.

NMI è il secondo come priorità; attivando questa linea, la CPU non si ferma subito, ma finisce l'operazione in corso, salva nello stack il Program Counter, e solo a questo punto il controllo viene passato alla routine puntata dal *vettore di interrupt*.

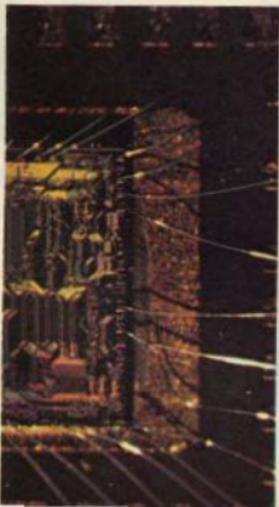
Quest'ultimo si trova allocato a 0066h (h sta per esadecimale).

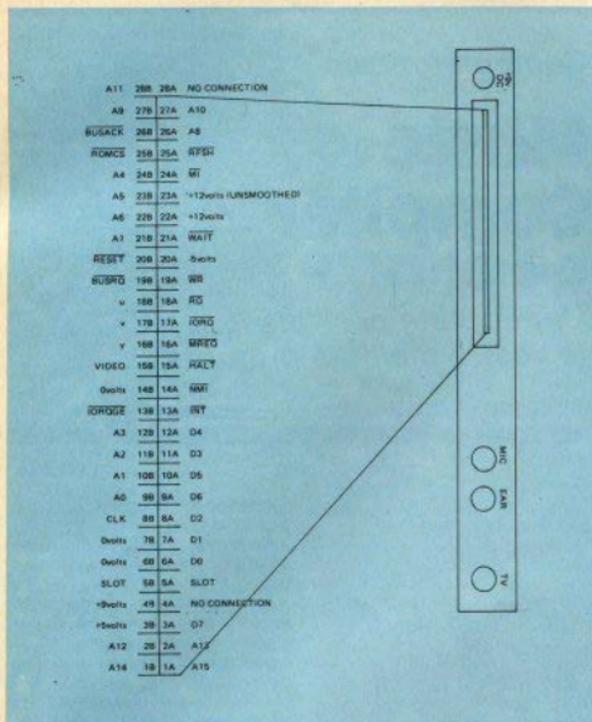
Conviene salvare nello Stack anche i registri e i flag prima di servire il dispositivo che ha richiesto l'interrupt, poiché durante lo svolgimento della routine possono essere modificati.

Prima di ritornare al programma principale, i dati si potranno prelevare inalterati dallo Stack.

INT ha la priorità più bassa ed è il più difficile da usare; infatti può essere usato in tre modi diversi. Nel primo modo, cioè come si trova predisposta la CPU dopo l'accensione o il *reset*, quando arriva la richiesta di *interrupt*, lo Z80 risponde con un segnale di IORQ e un segnale di M1, dopo il quale aspetterà che il dispositivo esterno metta un'istruzione sul bus dati entro il prossimo ciclo.

Il PC (Program Counter) viene salvato nello stack e si avvia la routine di gestione dell'*interrupt*; viene an-





che resettato un *flip-flop* che impedisce ad altri interrupt di essere esaminati: prima del ritorno al programma occorre perciò riabilitare il *flip-flop*.

Nel modo 2, ricevendo la richiesta di interrupt, la CPU si comporta come nel caso di NMI, salvo il fatto che il vettore di interrupt si trova a 0038h, e che l'interrupt stesso può essere mascherato, settando il *flip-flop*.

Nel modo 3, deve essere il dispositivo esterno a comunicare il vettore di interrupt alla CPU, fornendo 7 bit che, aggiunti agli 8 contenuti nello speciale registro I dello Z80, costituirà l'indirizzo di una tabella formata da 8 parole di 2 Byte. Ognuna di queste coppie è l'indirizzo di

una routine particolare, per la gestione dell'interrupt richiesto da una particolare periferica. Ogni INT può essere disattivato (mascherato), settando il *flip-flop* al quale ho accennato poco più sopra.

HALT è un segnale di conferma che proviene dalla CPU dopo aver eseguito l'istruzione HALT.

MREQ e IORQ sono segnali di riconoscimento emessi dalla CPU il primo conferma la validità degli indirizzi durante la lettura o la scrittura in memoria, il secondo indica che l'indirizzo presente sul bit 0 + 7 del Bus degli indirizzi è valido per l'I/O, permettendo la lettura o la scrittura di dati verso il dispositivo I/O interessato. L'IORQ è generato contemporaneamente a M1, quando lo Z80

riconosce un interrupt.

RD e WR sono rispettivamente il segnale di lettura ed il segnale di scrittura che la CPU emette durante queste operazioni. Indicano che la CPU è pronta a leggere o scrivere sia in memoria sia in qualche dispositivo esterno.

WAIT (attesa) è un ingresso della CPU che viene usato con una memoria o con una periferica lenta: quando è attivo, indica che il dispositivo non è ancora pronto al trasferimento dei dati. In questo caso la CPU continua a eseguire NOP, cioè non fa niente altro che generare il *refresh* della memoria.

M1 è un'uscita della CPU che indica che è in corso il primo ciclo macchina; poiché ogni operazione della CPU richiede alcuni cicli per essere completata, viene usata insieme a IORQ per sincronizzare i dispositivi esterni con la CPU stessa, quando viene attivato un interrupt.

RFSH (in uscita dalla CPU) è il segnale di rinfresco per la memoria; viene generato periodicamente con la giusta temporizzazione per operare il refresh con un minimo di hardware esterno.

RESET: portando a «0» per un momento questo ingresso, si reinizializza la CPU, in pratica è come se il computer venisse spento e nuovamente acceso (nel programmatore di EPROM ci sarà un apposito pulsante: è molto meglio che togliere e mettere lo spinotto in caso di «crash» del sistema).

ROMCS termina sul piedino 20 della ROM; se viene portato a +5V, la ROM viene disattivata. Questa operazione è necessaria, per esempio, per passare il controllo della CPU a una ROM o EPROM esterna, che conterrà i dati necessari.

BUSAK è il segnale di risposta della CPU quando, in seguito a un BUSRD, il bus degli indirizzi, quello dei dati e quello di controllo sono posti in three-state; comunica ai dispositivi esterni che è possibile lo scambio di dati con la memoria.

Gli altri segnali: Video, Y, V, U per il momento non ci interessano; restano le tensioni di 9, -12, e ben tre 0V. Alcune di queste tensioni verranno usate dal programmatore di EPROM,

Se avete copiato il lungo listato del mese scorso, questa volta con poco lavoro potrete aggiungere una routine che credo essere abbastanza originale e piacevole da utilizzare: non molte istruzioni che vi permetteranno di rappresentare graficamente qualche melodia.

Se vi è risultato chiaro quanto detto nel numero scorso e se non avete avuto problemi né difficoltà nell'inserire correttamente i dati, nessuna conoscenza ulteriore è richiesta per questo sottoprogramma. Anche l'uso è estremamente semplice, basta premere i tasti delle opzioni e il programma farà tutto il resto.

Naturalmente ricordatevi, prima di usarla, di aver caricato in macchina (con l'opzione INTRODUZIONE DATI o dal registratore) un qualche brano musicale.

Un altro consiglio: non inserire brani eccessivamente lunghi, la capienza dello schermo non è illimitata e incorrereste nell'errore *Integer Out of Range*. Mediamente è comunque possibile caricare 60/70 note.

Passiamo ora a vedere come utilizzarla: Dopo aver dato un RUN e aver caricato in macchina i dati relativi al pezzo di cui volete avere il grafico, dal menù principale, con l'opzione 2 (analisi dati in memoria) passate al menù secondario.

La scelta 6 vi manderà ad un altro menù secondario dal quale sono passibili 2 scelte:

- 1) Grafico linea melodica
- 2) Grafico linea ritmica.

La scelta 1 pulisce il video e dopo qualche secondo di attesa necessario per l'elaborazione dei dati, disegnerà una coppia di assi cartesiani.

In alto avrete le scritte:

MIN., φ MAX

dove min. indicherà la nota più bassa del pezzo (nella notazione numerica abituale del computer), max. la nota più alta e φ (lettera dell'alfabeto greco) l'intervallo in semitoni tra il quale è compreso il pezzo stesso.

Quindi sull'ordinata verranno disegnate delle linee orizzontali, in numero uguale al numero di semitoni.

Dopo questa preparazione del vi-

Computer music

Music Compiler

di Guido Grassi

Una routine per "disegnare" le vostre melodie preferite



deo, l'elaboratore passa alla stesura del pezzo vero e proprio.

Le note sono rappresentate con cerchietti neri posti sulle linee precedentemente tracciate, più o meno in alto, a seconda dell'altezza che il suono occupa all'interno della melodia, e quindi le unisce con delle linee, dando alla fine un grafico corri-

spondente all'andamento del brano musicale.

Terminata l'elaborazione, in fondo allo schermo compaiono le scritte: 1, 2, E, S.

Premendo 1 si ha la ripetizione delle operazioni appena compiute, premendo 2 si ha la rappresentazione, sempre in forma grafica, dell'andamento ritmico, premendo E è possibile avere l'esecuzione del pezzo e con S la stampa del video.

Scegliendo 2 si ottiene lo stesso risultato che si otterrebbe effettuando la scelta 2 del menù della routine "Graphic Melody".

Senza cancellare l'elaborazione precedente, il programma rappresenta graficamente la durata delle note. Anche qui i suoni vengono disegnati con cerchietti neri: quelli posti più in alto rappresentano le note di maggiore durata.

L'opzione E permette di ascoltare e "vedere" contemporaneamente il brano. Effettuata questa scelta, sulla parte bassa dello schermo compare la richiesta per il valore del metrone. Determinata la velocità desiderata, è possibile ascoltare e seguire sullo schermo, in una forma più immediata di quanto non permetta la tradizionale scrittura musicale, l'andamento melodico che si sta ascoltando.



Movimento dati

Sappiamo che l'istruzione LD permette di trasferire il contenuto di un registro a 8 bit in un altro registro a 8 bit, nella forma

LD r1, r2,
corrispondente a un'istruzione basic

LET r1 = r2.

Il dato da memorizzare in un registro può essere prelevato dalla memoria RAM, anziché da un altro registro, facendo uso della stessa istruzione LD; nel modo di indirizzamento immediato è possibile scrivere un valore specifico in ciascuno dei registri disponibili. Esempio:

LD E, 15

In questo caso il numero 15 viene caricato nel registro E. Poiché quest'ultimo è un registro a 8 bit, il numero specificato dovrà essere compreso fra 0 e 255 (00 e FF in esadecimale).

LD BC, 5566H

Questa istruzione carica il registro BC con il valore 5566 esadecimale, ovvero il registro B con il valore 55H ed il registro C con 66H.

Utilizzando il modo di indirizzamento diretto, solo i registri A, BC, DE, HL, SP, IX e IY possono essere specificati come sorgente o destinazione di un operando. Esempi:

LD A, (8000H)

Questa istruzione copia nel registro A il contenuto della locazione di memoria 8000H.

LD (ADDRESS), DE

Questa istruzione copia il contenuto del registro DE nella locazione di memoria specificata dalla 'label' ADDRESS.

Nel modo di indirizzamento indiretto vi sono altre limitazioni; qualsiasi singolo registro può essere usato come sorgente o destinazione del dato, purché l'indirizzo sia fornito indirettamente dal registro HL. Solo nel caso che il registro sorgente-destinazione sia A, è possibile usare BC o DE indirettamente. Esempi:

LD C, (HL)

Questa istruzione copia nel registro C il contenuto della locazione di memoria specificata dal registro HL.

LD (DE), A



Programmazione in assembly con lo Spectrum

a cura di Gianluca Carri

Nella puntata precedente (SC n. 11), a pag. 15, prima colonna, riga 12 dal basso, l'LD sorgente, destinazione e leggete LD destinazione, sorgente.

Questa istruzione copia nella locazione di memoria specificata dal registro DE il contenuto del registro A.

Volendo effettuare operazioni non previste dai modi di indirizzamento disponibili, sarà necessario ricorrere a opportune combinazioni di istruzioni. Esempi:

LD HL, 6000H

LD C, (HL)

In questo caso viene simulata un'istruzione LD C, (6000H) sfruttando HL nel modo di indirizzamento indiretto.

LD HL, (6000H)

LD (HL), E

Il registro E sarà memorizzato nella locazione di memoria il cui indirizzo è prelevato dalle locazioni 6000H e 6001H.

Lo scambio di dati fra due registri può essere effettuato con l'istruzione

LD, con PUSH/POP, oppure con la speciale istruzione EX DE, HL; che scambia i contenuti dei registri DE e HL. Esempi:

LD A, B

LD B, C

LD C, A

In questo caso i contenuti dei registri B e C sono scambiati facendo uso di A come memoria temporanea.

PUSH BC

PUSH HL

POP BC

POP HL

Lo scambio effettuato fra i registri HL e BC è ottenuto prelevando i registri dallo stack in ordine inverso. Notate come l'esempio che segue sia più efficiente:

PUSH BC

EX (SP), HL

POP BC

Operazioni aritmetiche

Le istruzioni ADD e SUB non permettono l'indirizzamento diretto, ovvero non è possibile eseguire un'istruzione come ADD A, (1000H). Un'ulteriore limitazione è rappresentata dal fatto che solo il registro A (e il registro HL per la sola ADD) può fornire il primo operando dell'operazione.

Esempi:

ADD, A, (HL)



Questa istruzione somma al registro A il contenuto della locazione di memoria indirizzata da HL.

ADD A, 10

In questo caso, la costante 10 è sommata al contenuto del registro A.

Combinando più istruzioni è possibile ottenere risultati non previsti dai modi di indirizzamento disponibili:

LD HL, (9000H)

LD BC, 100

ADD HL, BC

LD (9000H), HL

La costante 100 è sommata al contenuto della locazione di memoria 9000H.

L'istruzione CP è usata assai frequentemente per effettuare salti condizionati. In seguito a una sottrazione (A-B), i flags vengono modificati come segue:

flag zero settato se A = B

flag carry settato se A minore di B

Le seguenti istruzioni effettuano un salto a FINE se A è maggiore o uguale a 12:

CP 12

JR NC, FINE

...

FINE ...

L'istruzione NEG effettua la negazione del registro A, ovvero $A = 0 - A$. Se è necessario effettuare la negazione di un registro diverso da A, è possibile usare la sequenza:

SUB A

SUB registro

LD registro, A

Le istruzioni INC e DEC sono estremamente utili per aggiungere o sottrarre la costante 1 a un registro o a una locazione di memoria, indirizzata indirettamente da HL o da un registro indice. Esempi:

INC HL

corrisponde a $LET HL = HL + 1$

LD HL, 7800H

DEC (HL)

il contenuto della locazione di memoria 7800H è decrementato di 1.

Notate che un'istruzione come INC A non modifica in nessun caso il flag carry; se ciò fosse necessario, è possibile usare ADD A, 1.

Auto data

di Giovanni Mellina

È un programma di utilità per Spectrum, un pò particolare: non si crea... ma un pò si distrugge ...e un pò si trasforma.

Lo scopo originale è quello di ottenere, eseguendolo, un programma autonomo, predisposto per il caricamento del set di *user graphics*, con statements DATA autogenerati e corrispondenti agli UDG al momento presenti in memoria.

I DATA così ottenuti possono poi essere "MERGEati" con altri programmi che li utilizzano.

Note di programma:

— linea 9120: la variabile *i* viene impostata con l'indirizzo del primo statement DATA, in linea 9001.

— linee 9130:9170: il contenuto dei bytes di memoria letti viene registrato negli statements DATA.

— linee 9180:9200: la variabile *j*, ora contenente l'indirizzo della linea successiva all'ultimo DATA, viene memorizzata nella variabile di sistema VARS; per conseguenza, il successivo comando RUN 9999 causa la cancellazione delle linee 9100:9200, lasciando in memoria, come risultato finale, il solo programma di caricamento del DATA (linee 9000:9021), che può essere eseguito, salvato, o fuso con altro programma.

Lo scopo originale può essere facilmente esteso: molto semplicemente, il programma può essere adattato a convertire in DATA aree di memoria contenenti routines in codice macchina.

Ecco come:

— linea 9000: le variabili *F* e *T* dovranno essere impostate, rispettiva-

mente, con gli indirizzi di inizio e fine dell'area di memoria dove i DATA dovranno essere caricati.

— linee 9001:9099: devono essere creati tanti statements DATA, nell'esatto formato originale, quanti ne occorrono. Ciascuno conterrà l'equivalente di 8 bytes di memoria.

— linea 9130: il valore TO del ciclo FOR *x* deve corrispondere al numero di stat. DATA - 1.

— linea 9140: il valore originario 32600 deve essere sostituito con l'indirizzo di inizio dell'area di memoria da cui si vogliono prelevare i dati.

Importante: dopo aver digitato il programma, è necessario salvarlo su cassetta prima di eseguirlo, evitando la sgradita sorpresa di ritrovarlo completamente modificato.

Il programma può essere salvato con autostart LINE 9100 ed eseguito con GO TO 9100 (non con RUN!).

(list a pag. 41)

Billy

di Luca Ghisleni

Semplice ma ben realizzato giuoco a tempo in cui bisogna guidare il coniglietto Billy a «ripulire» un campo di carote. La difficoltà sta nel controllo di Billy, che continua a muoversi in direzione dell'ultimo tasto premuto (vengono usati i tasti direzionali 5,6,7,8 con le frecce corrispondenti).

Ogni volta che Billy attraversa un'area sgombra di carote, viene fatto bersaglio dei colpi di un cacciatore nascosto nel buio e, se viene colpito, lascia cadere una carota, con conseguente perdita di tempo. Buon divertimento.



Un complemento al programma Totocalcio di A. Russo, pubblicato sui numeri 4 e 5. È utilizzabile anche fornendo semplicemente l'array con le colonne [C\$(1500, 13)] e il numero delle colonne da stampare ("col").

I possessori della stampante SEIKOSHA GP50-S potranno stampare direttamente SULLE SCHEDE del Totocalcio le colonne dei sistemi ridotti, semplicemente inserendo la schedina nella stampante (togliete il rotolo di carta, prima...!), seguendo due semplici regole:

1) La schedina deve essere inserita allineata a destra del portacarte della stampante, dal lato minore della parte dove sono stampate le partite e, naturalmente, con la parte da stampare rivolta verso la testina. Fate attenzione a inserirla ben allineata.

2) Guardando una schedina, potete vedere che la zona dove sono stampati i nomi delle squadre è delimitata da un rettangolo, di cui andrà considerato il lato destro, presso i grossi tratti neri orizzontali; questa riga sarà orizzontale con schedina inserita correttamente nella stampante.

Dove far coincidere tale linea con il margine superiore del nastro della stampante (regolarsi a sinistra, perché il nastro della GP50S si abbassa leggermente sulla destra).

Con pochi minuti di pratica, vedrete che sono regole veramente elementari e immediate.

Il metodo seguito è molto semplice: si stampa sul video la «griglia» di quadratini in una opportuna disposizione, e quindi si fa un COPY.

La "finestra" di stampa è posizionata a partire dal pixel 64 orizz. e 13 vert., e occupa 175 per 143 pixel; i quadratini sono di 7 per 7 pixel e sono ottenuti con delle DRAW, non con un carattere grafico, per poter ottenere spostamenti della stampa anche di singoli pixel.

Il menu contiene 6 opzioni: con la prima si ottiene la stampa su tutte le caselle della schedina, utile per controllare l'allineamento; la seconda opzione è quella per la stampa delle colonne (da usare quando la schedina sarà posizionata in modo ottimale); la terza è per lo sposta-

Totoseikosha

di Fabio Berno

CONCORSO 26
COMITATO REGIONALE NAZIONALE ITALIANO
PARTITE DEL 24-2-85

Totocalcio
"AL SERVIZIO DELLO SPORT"

Squadra 1ª	Squadra 2ª	1	2	3	4
1 Ascoli	Avellino	(X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
2 Cremonese	Atalanta	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
3 Fiorentina	Sampdoria	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
4 Inter	Torino	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
5 Juventus	Verona	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
6 Napoli	Lazio	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
7 Roma	Milan	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
8 Udinese	Como	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
9 Arezzo	Triestina	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
10 Lecce	Bari	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
11 Pescara	Perugia	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
12 Salernitana	Palermo	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)
13 Foligno	Civitavecchia	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)	(1) (X) (2)

DIAPOSITIVE A COLORI
3M Film COLO SLIDE

(list a pag. 33)

mento di un certo numero di pixel della finestra di stampa (attenzione a non uscire dai limiti dello schermo!); la quarta riporta la finestra in posizione standard, ed è utile in caso di eventuali pasticci fatti usando la terza opzione; la quarta e la quinta sono per il salvataggio e il caricamento dell'array che contiene le colonne del sistema ridotto; la sesta ferma il programma o, nel caso si sia all'interno di "Totocalcio" di A. Russo, ritorna al menu principale.

I listati, come vedete, sono due: quello lungo, numerato a partire da

2100, è utilizzabile da solo (fornendo le colonne e il loro numero) o, "MERGEandolo", con il programma "Totocalcio".

Il secondo listato è da usare solo con il nostro "Totocalcio", insieme al programma di stampa, e va MERGEato dopo aver caricato "Totocalcio" mixato con la nuova porzione.

Riassunto, la procedura è:

- 1) LOAD "Totocalcio": Stop
- 2) MERGE "" [1° listato, che parte da 2100]
- 3) MERGE "" [2° listato]

Ho fatto 13?

di Carlo Padricello

(list a pag. 32)

Questo programma è rivolto ai giocatori del Totocalcio che vogliono seguire minuto per minuto la situazione-punti delle loro schedine.

È infatti possibile aggiornare i risultati delle partite in schedina e controllare immediatamente la situazione delle colonne o dei sistemi integrali giocati.

Una limitazione può essere il limitato numero di colonne che si possono inserire, cioè 20: ma chi non ha problemi di occupazione di memoria (cioè chi ha la versione a 48K) può alterare a piacere il "20" alla linea 48, ricordandosi però che so' facendo deve gestire lo schermo in modo che prima visualizzi le prime 20 colonne, poi le successive 20 e così via; questo è realizzabile alterando alcune linee come segue:

```
3025 FOR z = 1 TO nc: FOR k = 1
TO 20: ...il resto va lasciato uguale
3070 NEXT k: PRINT 0, "preme-
re un tasto per passare alle colonne
successive": PAUSE 0: FOR i = 1 TO
13: PRINT AT 1+i, 11; "[20 spazi]":
NEXT i: NEXT z
```

Il programma contiene ampie spiegazioni-guida per l'utente ed è semplice da usare: buona fortuna.

Joystick

di Raffaele Ilardo

(list a pag. 36)

Il joystick (e relativa interfaccia) prodotto dalla Kempston è senza

dubbio uno dei più diffusi, e numerosi sono anche i tipi di joystick Kempston-compatibili, che usano lo stesso sistema di input.

Tuttavia non tutti conoscono il modo di usare tale periferica da programma, poiché il sistema Kempston opera su porte logiche d'ingresso il cui indirizzo non è facile da trovare. Un indirizzo utile è il 44244: leggendo da Basic un byte da questa porta con la funzione (cioè facendo ad esempio LET J = IN 44244) si vede che in corrispondenza delle diverse posizioni del joystick si leggono valori diversi, che quindi possono essere usati come meglio si crede nel nostro programma. Naturalmente essendo in ambiente basic la velocità non sarà esaltante.

A questo problema rimedia una breve routine in ling. macchina, mediante la quale è possibile ottenere il movimento voluto sul video semplicemente leggendo con delle PEEK le locazioni di memoria 23549, 23550 e 23551, che contengono rispettivamente il flag del tasto FIRE (0 = non premuto, 1 = sì), e la linea e la colonna attuale di stampa.

Così la stampa si ottiene semplicemente con un ciclo:

```
10 PRINT AT PEEK 23550, PEEK
23551; "...
15 IF (PEEK 23459 = 1) THEN ...
20 RANDOMIZE USR 23456: GO
TO 10
```

Da notare che la routine è completamente rilocabile: qui è collocata nell'area del buffer stampante, ma è consigliabile spostarla oltre RAMTOP, naturalmente usando valori opportunamente diversi da 23549, 23550 e 23551, a cui andrà sommata una "costante di rilocazione" (lo spostamento).

La routine viene caricata in memoria dall'indirizzo 23456 e occupa 96 byte; le linee di caricamento sono quelle da 1 a 6, le altre sono i 2 esempi. È possibile ridurre il campo degli spostamenti VERTICALI modificando il terzultimo dato della linea 3 (è un 22), e di quelli ORIZZONTALI, agendo sul nono dato della linea 4 (è un 32).

Battaglia

by CIASPE

(lista pag. 27)

Il gioco si richiama alla classica battaglia navale, ma è notevolmente più complesso.

Trattandosi di sottomarini, non si manovra solo sul piano, ma anche in profondità: per ottenere questo, la classica griglia con le coordinate della battaglia navale (qui di 7x7 caselle) è triplicata: avete tre campi (mari) di battaglia, uno sopra l'altro, per un totale di 147 caselle, il livello di profondità è in ogni momento visibile sul monitor, insieme al numero di sottomarini da sistemare.

La dislocazione di questi avviene nel modo consueto, con i tasti cursore; con «L» si cambia livello, CON ENTER si assegna la posizione al «pezzo».

Nel fare questo, le caselle adiacenti al sottomarino piazzato vengono barrate, a ricordare che in esse non si possono mettere altri sottomarini; ugualmente vengono barrate, come potete verificare cambiando livello, le caselle in verticale con quella scelta.

Finita la sistemazione (i pezzi disponibili sono 15), occorre attendere circa 20 secondi, per dar modo al computer di collocare i suoi. Quindi si gioca, con tre colpi consecutivi a testa, da sparare con le solite coordinate, dopo aver selezionato il livello.

Premendo SPACE si visualizza la situazione (mezzi ancora in gioco); questa avviene automaticamente a ogni affondamento, dopo il quale si rientra nel gioco premendo un tasto qualsiasi.

Se vince lo Spectrum, potete alla fine andare a vedere dove aveva piazzato i suoi sottomarini.



Ecco un ottimo programma per visualizzare i dati nella forma di istogrammi, sia orizzontali che verticali, oppure di grafici di «spezzate» interpolanti i dati stessi.

Particolarità interessante del programma è che i comandi a disposizione dell'utente sono tutti presenti in alto sullo schermo, e questo è stato reso possibile da una routine grafica per la stampa di caratteri su 64 colonne.

I codici dei caratteri per la stampa a 64 colonne sono da caricare a partire dall'indirizzo 64256 e sono 800; dopo averli immessi, salvateli su nastro, in modo che il programma principale possa caricarli, come indicato nel listato. La routine che provvede alla stampa a 64 caratteri è quella tra le linee 650 e 690 incluse (questo a beneficio di chi proverà a usarla anche per altre realizzazioni), e il suo funzionamento è molto semplice da capire: i caratteri per la stampa a 64 colonne devono per forza occupare 4 bit in luogo degli 8 bit dei caratteri normali, quindi i codici memorizzati si riferiscono tutti a ipotetiche "celle" a 4 bit ed è per questo che il valore di ogni singolo codice non può superare il valore decimale 15 (= $8 + 4 + 2 + 1$).

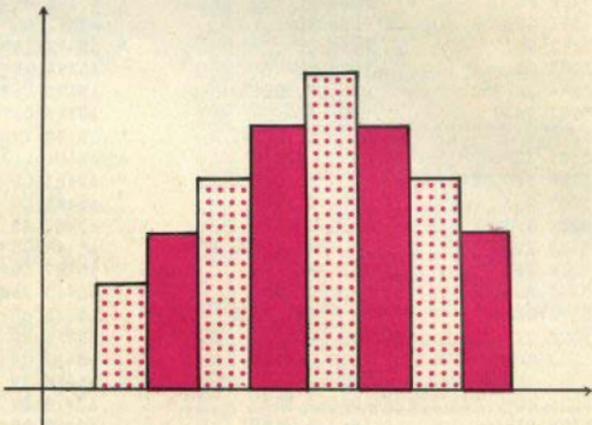
Ricordando inoltre come lo SPECTRUM gestisce i caratteri per il video (matrici 8 per 8 di punti, "descritte" da 8 codici, che si riferiscono alla configurazione dei punti delle 8 linee del carattere, codici ottenuti ognuno dal numero binario che ha gli "1" al posto dei puntini "accesi" sul video e gli "0" al posto di quelli "spenti" sulla linea considerata, con i bit più significativi a sinistra), si vede subito che per riempire un carattere normale servono 2 mini-caratteri, e che il primo di essi, cioè quello a sinistra, dovrà avere i suoi codici a 4 bit nelle posizioni più significative della matrice 8 per 8, quindi i codici delle configurazioni a 4 bit dovranno "scorrere" a sinistra di 4 posizioni: questo, in aritmetica binaria, significa che vanno moltiplicati per $2^4 = 16$.

A questo nuovo codice va poi aggiunto quello del mini-carattere che si posizionerà a destra della matrice, quindi sui 4 bit meno significativi, e per questo manterrà semplice-

Grafici

di Marco Manunta

(list a pag. 34)



mente il suo codice, che verrà sommato a quello (moltiplicato per 16) dell'altro carattere.

La nuova matrice 8 per 8 è così pronta, e viene inviata al video sotto forma di UDG, stampando cioè il carattere grafico ottenuto POKANDO gli 8 codici ottenuti come si fa per qualsiasi carattere grafico.

Detto questo, per quanto riguarda il funzionamento del programma bisogna dire che l'utente è sempre guidato in modo da non sbagliare, e che è possibile svolgere le operazioni seguenti semplicemente premendo il tasto corrispondente:

"X": arresto dell'elaborazione.

"D": hard-copy del video su stampante.

"I": INPUT dei valori: è possibile interromperlo in ogni momento inserendo "f" invece di un numero.

"A" o "B": si seleziona il numero di colonne dell'istogramma, (12 o 24 per l'istogramma verticale e 6 o 12 per quello verticale).

"H": rappresentazione a barre orizzontali (6 o 12)

"V": rappresentazione a barre verticali (12 o 24)

"G": rappresentazione a grafico cartesiano.

"C": modifica delle didascalie (ognuna di 3 caratteri).



Orologio

65276 15	LFEFC	DEC	D	65366 C33800	JP	00056
65277 3615		LD	(HL),#15	65369 7E	LFF59	LD A,(HL)
65279 0A		LD	A,(BC)	65370 3C		INC A
65280 FF		RST	#38	65371 B9		CP C
65281 F3		DI		65372 2001		JR NZ,LFF5F
65282 3EFE		LD	A,#FE	65374 AF		XOR A
65284 ED47		LD	I,A	65375 77	LFF5F	LD (HL),A
65286 ED5E		IM	2	65376 C9		RET
65288 FB		EI		65377 0600	LFF61	LD B,#00
65289 C9		RET		65379 4E		LD C,(HL)
65290 F5		PUSH	AF	65380 79		LD A,C
65291 C5		PUSH	BC	65381 D60A	LFF65	SUB #0A
65292 D5		PUSH	DE	65383 E600		AND #00
65293 E5		PUSH	HL	65385 2007		JR NZ,LFF72
65294 3A785C		LD	A,(23672)	65387 79		LD A,C
65297 D632		SUB	#32	65388 D60A		SUB #0A
65299 203D		JR	NZ,LFF52	65390 04		INC B
65301 32785C		LD	(23672),A	65391 4F		LD C,A
65304 21FCFE		LD	HL,LFEFC	65392 18F3		JR LFF65
65307 0E3C		LD	C,#3C	65394 78	LFF72	LD A,B
65309 CD59FF		CALL	LFF59	65395 C630		ADD A,#30
65312 200C		JR	NZ,LFF2E	65397 CD7FFF		CALL LFF7F
65314 23		INC	HL	65400 79		LD A,C
65315 CD59FF		CALL	LFF59	65401 C630		ADD A,#30
65318 2006		JR	NZ,LFF2E	65403 CD7FFF		CALL LFF7F
65320 23		INC	HL	65406 C9		RET
65321 0E10		LD	C,#10	65407 E5	LFF7F	PUSH HL
65323 CD59FF		CALL	LFF59	65408 C5		PUSH BC
65326 3EF7	LFF2E	LD	A,#F7	65409 ED4B365C		LD BC,(23606)
65328 DBFE		IN	A,(#FE)	65413 2600		LD H,#00
65330 E600		AND	#00	65415 6F		LD L,A
65332 C252FF		JP	NZ,LFF52	65416 29		ADD HL,HL
65335 111840		LD	DE,16400	65417 29		ADD HL,HL
65338 21FEFE		LD	HL,LFEFE	65418 29		ADD HL,HL
65341 CD61FF		CALL	LFF61	65419 09		ADD HL,BC
65344 3E3A		LD	A,#3A	65420 D5		PUSH DE
65346 CD7FFF		CALL	LFF7F	65421 0600		LD B,#00
65349 2B		DEC	HL	65423 7E	LFF8F	LD A,(HL)
65350 CD61FF		CALL	LFF61	65424 12		LD (DE),A
65353 3E3A		LD	A,#3A	65425 14		INC D
65355 CD7FFF		CALL	LFF7F	65426 23		INC HL
65358 2B		DEC	HL	65427 10FA		DJNZ LFF8F
65359 CD61FF		CALL	LFF61	65429 D1		POP DE
65362 E1	LFF52	POP	HL	65430 1C		INC E
65363 D1		POP	DE	65431 C1		POP BC
65364 C1		POP	BC	65432 E1		POP HL
65365 F1		POP	AF	65433 C9		RET
				65434 F3		DI
				65435 ED56		IM 1
				65437 FB		EI
				65438 C9		RET
				65439 00		NOP
				65440 00		NOP



Battaglia

```

1: LET y1=1
825 PRINT AT 18,17;"QUANDO SEI"
;AT 19,17;"POSIZIONATO";AT 20,17
;"PREMI ";FLASH 1;"ENTER"
830 PRINT AT 18,2;"LIVELLO: ";1
1
840 PRINT AT 20,2;"SOMM. ";;sb
850 PRINT AT y1*2,0; INK 7; PAP
ER 0;y1
860 PRINT AT 0,x1*2; INK 7; PAP
ER 0;CHR$(64+x1)
865 PRINT PAPER 0; INK 7;AT y1
*2,x1*2; OVER 1;" "
866 REM *****
867 REM * MOVIMENTO CURSORE *
868 REM *****
869 REM
870 IF INKEY$="5" AND x1>1 THEN
LET x1=x1-1: GO SUB 950
880 IF INKEY$="8" AND x1<7 THEN
LET x1=x1+1: GO SUB 950
890 IF INKEY$="7" AND y1>1 THEN
LET y1=y1-1: GO SUB 950
900 IF INKEY$="6" AND y1<7 THEN
LET y1=y1+1: GO SUB 950
910 IF INKEY$="L" THEN LET 11=
11+1: GO SUB 1500
920 IF INKEY$=CHR$ 13 THEN GO
SUB 1000
930 GO TO 850
950 REM *****
951 REM * ROUTINE AZZERAMENTO *
952 REM * RIFERIMENTI *
953 REM *****
954 REM *****
955 REM *****
956 REM
960 PRINT AT a(1)*2,0;a(1)
965 PRINT AT 0,a(2)*2;CHR$(64+
a(2))
967 PRINT BRIGHT 1; PAPER 5;AT
a(1)*2,a(2)*2;a$(a(1),a(2),11)
970 LET a(1)=y1: LET a(2)=x1
975 IF sb=0 THEN GO TO 1600
980 RETURN
1000 REM *****
1003 REM * ROUTINE RIEMPIMENTO *
1005 REM * CASELLE ADIACENTI *
1007 REM * IL SOMMERSIBILE *
1010 REM *****
1011 REM
1020 IF a$(y1,x1,11)=" " THEN L
ET a$(y1,x1+1,11)=CHR$ 162: PRINT
BRIGHT 1;AT y1*2,x1*2;CHR$ 162:
GO TO 1025
1022 BEEP 1,20: RETURN
1025 LET sb=sb-1
1026 PRINT AT 20,11-LEN STR$ sb;
";sb
1030 FOR b=1 TO 3: IF a$(y1,x1,b
)=" " THEN LET a$(y1,x1,b)="#"
1035 NEXT b
1040 FOR n=y1-1 TO y1+1
1042 IF n=0 OR n=8 THEN GO TO 1
060
1045 FOR h=x1-1 TO x1+1
1047 IF h=0 OR h=8 THEN GO TO 1
055
1050 IF a$(n,h,11)=" " THEN LET
a$(n,h,11)="#"
1055 NEXT h
1060 NEXT n
1065 IF sb=0 THEN FOR n=18 TO 2

```



```

1: PRINT AT n,17;"
  : NEXT n: PRINT AT 18,17;"TUTT
  I I TUOI":AT 19,17;"SOMMERGIBILI
  ":AT 20,17;"SONO A POSTO"; FLASH
  0: FOR n=1 TO 2: BEEP .4,10*n:
NEXT n: GO SUB 950
1070 GO SUB 1500
1080 RETURN
1100 REM *****
1103 REM * SISTEMAZIONE *
1105 REM * SOMMERGIBILI *
1107 REM * SPECTRUM *
1110 REM *****
1111 REM
1120 FOR n=18 TO 21: PRINT AT n,
17;"
1125 BEEP .3,50: PRINT AT 18,17;
"ORA SISTEMO":AT 19,17;"I MIEI";
AT 20,17;"SOMMERGIBILI";
1130 REM PRINT AT 17,16;"

```

```

1150 LET xx=INT (RND*7)+1: LET y
y=INT (RND*7)+1: LET ll=INT (RND
*3)+1
1160 IF b$(yy,xx,ll)=" " THEN L
ET b$(yy,xx,ll)=CHR$ 163: GO SUB
1200
1180 GO TO 1150
1200 REM *****
1203 REM * ROUTINE RIEMPIMENTO *
1204 REM * CASELLE ADIACENTI *
1206 REM * SOTTOMARINI ZX *
1210 REM *****
1211 REM
1215 LET sc=sc-1: IF sc=0 THEN
GO TO 1600
1220 FOR n=yy-1 TO yy+1
1230 IF n=0 OR n=8 THEN GO TO 1
265
1235 FOR h=xx-1 TO xx+1
1240 IF h=0 OR h=8 THEN GO TO 1
260
1250 IF b$(n,h,ll)="#" THEN LET
b$(n,h,ll)="#"
1260 NEXT h
1265 NEXT n
1270 FOR b=1 TO 3: IF b$(yy,xx,b
)=" " THEN LET b$(yy,xx,b)="#"
1280 NEXT b
1290 RETURN
1500 REM *****
1505 REM * VISUALIZZAZIONE *
1507 REM * SOTTOMARINI *
1509 REM * GIOCATORE *
1510 REM *****
1511 REM
1515 IF ll=4 THEN LET ll=1
1518 PRINT AT 18,11,ll
1520 FOR o=1 TO 7: FOR p=1 TO 7
1530 PRINT BRIGHT 1; PAPER 5;AT
o*2,p*2;a$(o,p,ll)
1540 NEXT p: NEXT o
1545 IF sb=0 THEN GO TO 1100
1550 RETURN
1600 REM *****
1603 REM * PULIZIA MATRICI *
1605 REM * DI GIOCO *
1610 REM *****
1611 REM
1620 FOR b=1 TO 3: FOR g=1 TO 7:
FOR h=1 TO 7

```

```

1630 IF a$(g,h,b)="#" THEN LET
a$(g,h,b)="#"
1635 IF b$(g,h,b)="#" THEN LET
b$(g,h,b)="#"
1640 NEXT h: NEXT g: NEXT b
1645 IF sc=0 THEN GO TO 1800
1650 LET ll=1: GO SUB 1500
1800 REM *****
1805 REM * INIZIO PARTITA *
1810 REM *****
1811 REM
1815 FOR n=17 TO 21: PRINT AT n,
0;"
": NEXT n
1816 BEEP .5,40
1817 PRINT AT 19,0;"POSSIAMO INI
ZIARE":AT 21,19;"press any key"
1818 IF INKEY$=" " THEN GO TO 18
18
1820 FOR n=17 TO 21: PRINT AT n,
0;"
": NEXT n
1822 REM *****
1823 REM * MAPPA DI GIOCO *
1826 REM *****
1827 REM
1830 FOR n=28 TO 44 STEP 16: PLO
T 12,n: DRAW 88,0: PLOT 140,n: D
RAW 88,0
1835 PLOT 12,n+1: DRAW 88,0: PLO
T 140,n-1: DRAW 88,0: NEXT n
1840 FOR n=0 TO 1: PLOT 12+88*n,
28: DRAW 0,16: PLOT 140+88*n,28:
DRAW 0,16: NEXT n
1845 PRINT AT 17,2;"LIVELLO: ";1
1:AT 17,18;"LIVELLO: ";12
1850 FOR n=3 TO 21 STEP 16: PLOT
4,n: DRAW 248,0
1855 PLOT 4,n+1: DRAW 248,0: NEX
T n
1860 FOR n=4 TO 252 STEP 248: PL
OT n,4: DRAW 0,14: NEXT n
1863 PLOT 32,4: DRAW 0,14: PLOT
122,4: DRAW 0,14
1865 IF u1=1 THEN RETURN
1866 LET u1=1
1867 LET sb=15: LET sc=15
1870 PRINT PAPER 2; INK 7; FLAS
H 1;AT 20,1;"TU"; FLASH 0; PAPER
4; INK 0;AT 20,5;"COLPI: ";AT 20
,7;"POSIZIONE: ";
1873 PRINT AT 20,11;" ";AT 20,28
"
1875 PLOT 32,4: DRAW 0,14: PLOT
122,4: DRAW 0,14
1877 PRINT AT 20,12;cp
1880 REM *****
1885 REM * SCELTA PER *
1886 REM * CAMBIO DI LIVELLO E *
1887 REM * SITUAZIONE DI GIOCO *
1890 REM *****
1891 REM
1900 IF INKEY$="" THEN GO TO 19
00
1905 IF INKEY$="P" THEN LET ll=
ll+1: BEEP .4,-20: GO SUB 2600
1910 IF INKEY$="L" THEN GO TO 2
500
1912 IF INKEY$="" THEN LET p1=
0: GO SUB 3000
1915 IF INKEY$=" " STOP " THEN ST

```



```

OP
1918 IF INKEY$="" THEN GO TO 19
00
1920 LET d$=INKEY$: LET d=CODE d
$
1924 REM *****
1925 REM * INPUT DA TASTIERA *
1926 REM * CASELLA DI TIRO *
1927 REM *****
1928 REM
1930 IF d<49 OR d>55 AND d<65 OR
d>71 THEN BEEP .4,0: GO TO 190
0
1935 IF r=1 AND d<65 OR d>71 THE
N BEEP .2,60: GO TO 1900
1940 IF r=1 THEN LET k$(1)=d$:
PRINT AT 20,28;d$: "-": PRINT AT
0,16+(d-64)*2: I< 7: PAPER 0;d$:
GO TO 1960
1945 IF r=2 AND d<49 OR d>55 THE
N BEEP .2,60: GO TO 1900
1950 IF r=2 THEN LET k$(2)=d$:
PRINT AT 20,30;d$: PRINT AT (d-4
8)*2,16: INK 7: PAPER 0;d$
1960 LET r=r+1: IF r=3 THEN GO
TO 2000
1970 GO TO 1900
2000 LET yy=CODE k$(2)-48: LET x
x=CODE k$(1)-64
2010 IF b$(yy,xx,12)=CHR$ 163 TH
EN LET p1=1: LET sc=sc-1: GO SU
B 3000
2020 IF b$(yy,xx,12)<>CHR$ 148 T
HEN LET b$(yy,xx,12)=CHR$ 151:
PRINT PAPER 6: BRIGHT 1;AT yy*
2,16+xx*2;CHR$ 151
2030 LET cp=cp-1: IF cp=0 THEN
GO TO 2040
2035 LET r=1
2038 IF INKEY$="" THEN GO TO 20
38
2040 PRINT AT 0,16+xx*2;k$(1)
2045 PRINT AT 20,28;" "
2050 PRINT AT yy*2,16;k$(2)
2053 IF cp=0 THEN LET cp=3: LET
r=1: GO TO 2200
2055 IF cp<>0 THEN PRINT AT 20,
12;cp
2060 GO TO 1900
2200 REM *****
2203 REM * ROUTINE DI FUOCO ZX *
2210 REM *****
2211 REM
2215 PRINT AT 20,12;cc;AT 20,28;
" "
2220 PRINT AT 20,1: FLASH 1;"ZX"
2250 FOR n=1 TO 100: NEXT n: LET
l1=INT (RND*3)+1
2255 IF l1<>11 THEN BEEP .4,-20
: LET l1=11: GO SUB 2600
2260 LET n1=0
2270 LET y1=INT (RND*7)+1: LET x
1=INT (RND*7)+1
2280 LET n1=n1+1: IF n1=20 THEN
GO TO 2250
2290 IF a$(y1,x1,11)=CHR$ 162 TH
EN GO TO 2297
2295 IF a$(y1,x1,11)<>" " THEN
GO TO 2270
2297 PRINT AT 20,28;CHR$(64+x1)
;"-";CHR$(48+y1): BEEP .02,40

```

```

2298 PRINT AT y1*2,0: OVER 1;CHR
$ 143;AT 0,x1*2: OVER 1;CHR$ 143
;AT y1*2,x1*2: INK 2;CHR$ 143
2299 IF a$(y1,x1,11)=CHR$ 162 TH
EN LET sb=sb-1: LET p1=2: GO SU
B 3000
2300 FOR n=1 TO 100: NEXT n
2302 IF a$(y1,x1,11)<>CHR$ 148 T
HEN LET a$(y1,x1,11)=CHR$ 151:
PRINT PAPER 5: BRIGHT 1;AT y1*2
,x1*2;CHR$ 151
2305 FOR n=1 TO 50: NEXT n: PRIN
T AT y1*2,0: OVER 1;CHR$ 143;AT
0,x1*2: OVER 1;CHR$ 143
2310 LET cc=cc-1: IF cc<>0 THEN
PRINT AT 20,12;cc: PRINT AT 20,
28;" "
2315 IF cc=0 THEN LET cc=3: FOR
n=1 TO 100: NEXT n: GO TO 1870
2320 GO TO 2250
2500 REM *****
2503 REM * ROUTINE PER CAMBIO *
2505 REM * DI LIVELLO CAMPO *
2506 REM * GIOCATORE *
2510 REM *****
2511 REM
2520 LET l2=l2+1
2530 IF l2=4 THEN LET l2=1
2535 PRINT AT 17,27: INK 7: PAPE
R 0;l2
2537 BEEP .4,-20
2540 FOR g=1 TO 7: FOR h=1 TO 7
2550 IF b$(g,h,12)=CHR$ 163 THEN
PRINT PAPER 6: BRIGHT 1;AT g*
2,16+h*2;" " : GO TO 2570
2560 PRINT PAPER 6: BRIGHT 1;AT
g*2,16+h*2;b$(g,h,12)
2570 NEXT h: NEXT g
2585 PRINT AT 17,27;12
2590 GO TO 1900
2600 REM *****
2603 REM * ROUTINE PER CAMBIO *
2605 REM * DI LIVELLO CAMPO ZX *
2610 REM *****
2611 REM
2612 IF l1=4 THEN LET l1=1
2615 PRINT AT 17,11: PAPER 0: IN
K 7;l1
2620 FOR g=1 TO 7: FOR h=1 TO 7
2630 IF a$(g,h,11)="" THEN PRI
NT PAPER 5: BRIGHT 1;AT g*2,h*2
;" " : GO TO 2650
2640 PRINT PAPER 5: BRIGHT 1;AT
g*2,h*2;a$(g,h,11)
2650 NEXT h: NEXT g
2655 PRINT AT 17,11;l1: FOR n=1
TO 50: NEXT n
2660 RETURN
3000 REM *****
3002 REM * VISUALIZZAZIONE *
3004 REM * SOTTOMARINO COLPITO *
3005 REM * E *
3006 REM * SITUAZIONE DI GIOCO *
3010 REM *****
3011 REM
3012 IF p1=1 THEN FOR n=1 TO 8:
PRINT PAPER 6: BRIGHT 1;AT yy*
2,16+xx*2;CHR$ 148: BEEP .02,20:
PRINT PAPER 6: BRIGHT 1;AT yy*
2,16+xx*2;" " : NEXT n
3014 IF p1=2 THEN FOR n=1 TO 8:

```

```

PRINT PAPER 5; BRIGHT 1; AT y1*
2,x1*2; CHR$ 148; BEEP .02,20; PR
INT PAPER 5; BRIGHT 1; AT y1*2,x
1*2; " "; NEXT n
3020 FOR n=16 TO 21: PRINT AT n,
0; "
"; NEXT n
3022 FOR n=36 TO 20 STEP -16: PL
OT 12,n; DRAW 110,0; PLOT 140,n;
DRAW 110,0
3023 PLOT 12,n+1; DRAW 110,0; PL
OT 140,n+1; DRAW 110,0; NEXT n
3025 FOR n=12 TO 128 STEP 55: PL
OT n,20; DRAW 0,16; PLOT 128+n,2
0; DRAW 0,16; NEXT n
3030 IF p1=0 THEN PRINT AT 20,2
; PAPER 0; INK 7;"SITUAZIONE": G
O TO 3050
3040 PRINT AT 20,3; FLASH 1; PAP
ER 2; INK 7;"DISTRUTTO"
3050 BEEP .5,30
3060 PRINT AT 18,2;"FLOTTA"; AT 1
8,18;"FLOTTA"
3080 PRINT AT 18,29-LEN STR$ sc;
sc; AT 18,13-LEN STR$ sb; sb
3090 PRINT AT 20,18;"press any k
ey"
3092 IF p1=2 THEN LET a$(y1,x1,
11)=CHR$ 148; PRINT PAPER 5; BR
IGHT 1; AT y1*2,x1*2; CHR$ 148
3094 IF p1=1 THEN LET b$(yy,xx,
12)=CHR$ 148; PRINT PAPER 6; BR
IGHT 1; AT yy*2,16+xx*2; CHR$ 148
3096 IF sc=0 THEN GO TO 3300
3098 IF sb=0 THEN GO TO 3400
3100 IF INKEY$="" THEN GO TO 31
00
3110 FOR n=17 TO 21: PRINT AT n,
0; "
"; NEXT n
3111 REM *****
3112 REM * NUOVA MAPPA *
3113 REM * DI GIOCO *
3114 REM *****
3115 GO SUB 1830
3120 IF p1=2 THEN GO TO 3150
3130 PRINT AT 20,1; FLASH 1; PAP
ER 2; INK 7;"TU"; FLASH 0; PAPER
4; INK 0; AT 20,5;"COLPI:"; AT 20
,17;"POSIZIONE:"
3135 PRINT AT 20,12; cp: IF p1=1
THEN PRINT AT 20,12; cp-1
3140 RETURN
3150 PRINT AT 20,1; FLASH 1;"ZX"
; FLASH 0; AT 20,5;"COLPI:"; AT 20
,17;"POSIZIONE:"
3160 GO SUB 3200
3170 RETURN
3180 REM *****
3182 REM * BLOCCO CASELLE *
3184 REM * ADIACENTI SOTTOMA- *
3186 REM * RINO COLPITO (ZX) *
3190 REM *****
3191 REM
3200 FOR n=y1-1 TO y1+1
3210 IF n=0 OR n=8 THEN GO TO 3
260
3220 FOR h=x1-1 TO x1+1
3230 IF h=0 OR h=8 THEN GO TO 3
250
3240 IF a$(n,h,11)=" " THEN LET

```

```

a$(n,h,11)="#"
3250 NEXT h
3260 NEXT n
3270 FOR b=1 TO 3: IF a$(y1,x1,b
)=" " THEN LET a$(y1,x1,b)="H"
3280 NEXT b
3290 RETURN
3300 REM *****
3303 REM * VITTORIA GIOCATORE *
3306 REM * E ROUTINE CHIUSURA *
3310 REM *****
3311 REM
3312 FOR n=1 TO 150: NEXT n
3315 FOR n=17 TO 21: PRINT AT n,
0; "
"; NEXT n
3320 PRINT AT 18,0; FLASH 1;"HAI
VINTO"; FLASH 0;" QUESTA PART
ITA"
3330 PAUSE 100
3340 FOR n=0 TO 16: PRINT AT n,1
6; "
"; NEXT n
3350 PRINT AT 2,18;"* SE VUOI"; A
T 4,18;"GIOCARRE ANCORA"; AT 6,18;
"PRESS "; FLASH 1;"ENTER"; AT 8,1
8; FLASH 0;"* PER FINIRE"; AT 10,
18;"PRESS "; FLASH 1;"STOP"
3360 IF INKEY$=CHR$ 13 THEN CLS
: GO TO 400
3370 IF INKEY$="A" THEN STOP
3380 GO TO 3360
3400 REM *****
3405 REM * ROUTINE VITTORIA ZX *
3410 REM *****
3411 REM
3415 FOR n=1 TO 150: NEXT n
3420 FOR n=17 TO 21: PRINT AT n,
0; "
"; NEXT n
3430 PRINT AT 19,0; FLASH 1;"HO
VINTO IO"; FLASH 0;" QUESTA PART
ITA"
3440 PAUSE 100
3450 FOR n=0 TO 16: PRINT AT n,0
; "
"; NEXT n
3460 PRINT AT 2,0;"* SE VUOI"; AT
4,0;"VEDERE IL MIO"; AT 6,0;"SCH
IERAMENTO"; AT 8,0;"PRESS "; FLAS
H 1;"ENTER"; FLASH 0; AT 10,0;"*
ALTRIMENTI"; AT 12,0;"press any k
ey"
3470 IF INKEY$="" THEN GO TO 34
70
3480 IF INKEY$=CHR$ 13 THEN GO
SUB 3600
3490 GO TO 3330
3600 REM *****
3603 REM * VISUALIZZAZIONE *
3605 REM * CAMPO GIOCO ZX *
3610 REM *****
3611 REM
3615 PRINT AT 19,0;"
";
3620 FOR b=1 TO 3
3625 PRINT AT 17,18;"LIVELLO: ";
b
3626 FOR n=28 TO 44 STEP 16: PLO
T 140,n; DRAW 88,0
3627 PLOT 140,n-1; DRAW 88,0; NE
XT n
3628 FOR n=0 TO n-1: PLOT 140+88*n

```



```

,28: DRAW 0,16: NEXT n
3630 FOR g=1 TO 7
3640 FOR h=1 TO 7
3650 PRINT PAPER 6; BRIGHT 1;AT
g*2,16+h*2;b$(g,h,b)
3660 NEXT h: NEXT g
3670 PRINT AT 21,16;"press any k
ey"
3680 IF INKEY$="" THEN GO TO 36
80
3685 PRINT AT 21,16;"
"
3690 NEXT b
3695 PRINT AT 21,0;"press any ke
y"
3700 IF INKEY$="" THEN GO TO 37
00
3710 CLS : RETURN
4000 REM *****
4003 REM * DEFINIZIONE *
4005 REM * CARATTERI UDG *
4010 REM *****
4011 REM
4020 FOR n=USR "h" TO USR "h"+7:
READ h: POKE n,h: NEXT n
4030 FOR n=USR "e" TO USR "e"+7:
READ e: POKE n,e: NEXT n
4040 FOR n=USR "s" TO USR "t"+7:
READ s: POKE n,s: NEXT n
4050 DATA 60,66,153,165,153,
66,60,60,24,153,255,255,153,24,6
0
4060 DATA 129,66,60,24,24,126,25
5,60,129,66,60,24,24,102,195,60
4070 RETURN

```

Sillabe

1 REM Divisione sillabica
di Luca Ghisleni

```

70 INPUT a$
100 LET h=LEN a$
110 IF h<=2 THEN GO TO 410
120 LET i=1
130 GO SUB 430
140 IF l=0 THEN GO TO 380
150 LET l=1
160 LET i=i+1
170 IF i>h THEN GO TO 410
180 IF a$(i)="h" THEN GO TO 16
0
190 GO SUB 430
200 IF l=0 THEN GO TO 290
210 IF l=2 OR l=2 THEN GO TO
250
220 LET j=i-1
230 GO SUB 530
240 GO TO 100
250 LET i=i+1
260 IF i>h THEN GO TO 410
270 LET l=1
280 GO TO 190
290 LET i=i+1

```

```

300 IF i>h THEN GO TO 410
310 GO SUB 430
320 IF l=0 THEN GO TO 350
330 LET j=i-2
340 GO TO 230
350 GO SUB 570
360 IF s=0 THEN GO TO 220
370 GO TO 330
380 LET i=i+1
390 IF i>h THEN GO TO 410
400 GO TO 130
410 PRINT a$
420 GO TO 70
430 REM vocale o consonante
440 LET l=a$(i)
450 IF l$="i" OR l$="u" THEN G
O TO 510
460 IF l$="a" OR l$="e" OR l$="
o" THEN GO TO 490
470 LET l=0
480 RETURN
490 LET l=1
500 RETURN
510 LET l=2
520 RETURN
530 REM stampa della sillaba
540 PRINT a$(1 TO j);"-";
550 LET a$a$(j+1 TO h)
560 RETURN
570 REM gruppo consonantico
580 LET l$a$(i-1)
590 LET m$a$(i)
600 IF l$=m$ THEN GO TO 660
610 IF l$="g" AND m$="n" THEN
GO TO 680
620 IF l$="l" OR l$="m" OR l$="
n" OR l$="r" OR l$="v" OR l$="z"
THEN GO TO 660
630 IF l$="s" OR m$="r" THEN G
O TO 680
640 IF l$="t" THEN GO TO 660
650 IF m$="h" OR m$="l" THEN G
O TO 680
660 LET s=0
670 RETURN
680 LET s=1
690 RETURN

```

Billy

5>REM Billy
di Luca Ghisleni

```

10 GO SUB 500
12 LET tm=10000: DEF FN t()=IN
T ((65536*PEEK 23674+256*PEEK 23
673+PEEK 23672)/50)
15 INPUT "Difficolta'? (1-5):
";d
16 IF d<1 OR d>5 THEN GO TO 1
5
18 LET di=110-20*d
100 PRINT AT 0,0;
101 FOR i=0 TO 703: PRINT INK
4;CHR$ 144;: NEXT i

```



```

105 LET x=15: LET y=21: LET xo=
0: LET yo=0: LET f=2: LET a=120:
LET b=174: LET co=0: POKE 23674
,0: POKE 23673,0: POKE 23672,0
110 FOR i=1 TO 703
120 IF INKEY$="8" THEN LET xo=
1: LET yo=0
121 IF INKEY$="5" THEN LET xo=
-1: LET yo=0
122 IF INKEY$="7" THEN LET yo=
-1: LET xo=0
123 IF INKEY$="6" THEN LET yo=
1: LET xo=0
125 LET oy=y: LET ox=x
130 LET y=y+yo*(0<y AND yo<0)
OR (y<21 AND yo>0): LET x=x+xo*
(0<x AND xo<0) OR (x<31 AND xo>
0)
135 IF ATTR (y,x)=70 THEN LET
f=2
140 PRINT AT y,x:CHR$ 145
145 PRINT AT oy,ox: " "
160 GO TO 201-f
199 LET f=0: LET co=co+1: IF ((
co>=d1) AND (x<>ox OR y<>oy)) TH
EN GO SUB 300
200 GO TO 120
201 OUT 254,0: OUT 254,16: NEXT
i
250 GO TO 400
300 LET r=x*8-a: LET s=((21-y)*
8-b): LET l=RND*r: LET m=RND*s
301 PLOT OVER i: INK 8;a,b: DR
AW OVER i: INK 8;1,m: BEEP .01,
36
302 PLOT OVER i: INK 8;a,b: DR
AW OVER i: INK 8;1,m
303 IF (r-1)<8 AND (s-m)<8 THEN
PRINT INK 5:AT oy,ox:CHR$ 144
: LET i=i-1
304 LET a=a+1: LET b=b+m: LET c
o=0: RETURN
400 CLS : LET t=FN t()
410 IF t<tm THEN PRINT AT 10,0
;"Bravo, hai realizzato il migli
or tempo:": LET tm=t
420 PRINT AT 18,0:"Tempo:":INT
(t/3600):TAB 9:"": LET t=t-36
00*INT (t/3600): PRINT INT (t/60
):TAB 12:"": LET t=t-60*INT (t
/60): PRINT t:TAB 15
430 INPUT "Un'altra partita? (s
/n)";a$: IF a$="s" THEN GO TO 1
5
440 STOP
500 FOR n=0 TO 15
510 READ a: POKE USR "a"+n,a: N
EXT n
520 DATA 0,24,60,102,102,60,24,
0,130,68,40,124,84,254,230,124
600 INK 6: PAPER 0: BORDER 0: B
RIGHT 1: CLS
610 PRINT AT 4,0:" Scopo del g
ioco e' aiutare il nostro amico
Billy il coniglio amangiare nott
etempo tutte le ca-rote del camp
o facendo attenzio-ne al contadi
no che, armato di fucile, tenta
di sparare a Billyogni volta qu
esti debba passare allo scoperto
."

```

```

620 PRINT "Se viene colpito il
povero Billydeve lasciar cadere
una delle carote raccolte perde
ndo tempo prezioso."
630 PRINT "Dunque buon diverti
mento e cercate di realizzare il
minor tempo possibile."
640 RETURN

```

Ho fatto 13?

5>REM HO FATTO 13 ?

di C. Padricello

```

10 DIM m(20): DIM n(5): PRINT
: PRINT "-----INPUT INFORMAZIO
NI-----": PRINT
15 PRINT "CONCORSO N.": INPUT
con: PRINT con
20 DIM a$(13,2,10): FOR k=1 TO
13: INPUT (k): "incontro" "1a s
quadra ";a$(k,1) "2a squadra ";a
$(k,2)
25 PRINT AT k+4,0;k:AT k+4,3;a
$(k,1) "-" ;a$(k,2)
30 NEXT k: PRINT AT 20,0:"COMF
ERMI I DATI INSERITI ?": INPUT "
s/n " ,h$: IF h$="n" THEN CLS :
GO TO 20
35 LET nc=0: LET ns=0: CLS : P
RINT "RISPONDI CON s/n"
40 PRINT AT 2,0:"Vuoi inserire
colonne singole? " : INPUT h$:
IF h$<"s" AND h$<"n" THEN GO
TO 40
45 PRINT h$: IF h$="n" THEN G
O TO 50
46 PRINT "Quante colonne? " :
48 INPUT nc: IF nc<1 OR nc>20
THEN GO TO 48
49 PRINT nc: DIM c$(nc,13)
50 PRINT AT 4,0:"Vuoi inserire
sistemi? " : INPUT h$: IF h$<"
s" AND h$<"n" THEN GO TO 50
55 PRINT h$: IF h$="n" THEN G
O TO 65
57 PRINT "Quanti sistemi? " :
60 INPUT ns: IF ns>5 THEN GO
TO 59
62 PRINT ns: DIM s$(ns,13,3)
65 CLS : PRINT "CONCORSO TOTOC
ALCIO N.":con
70 IF nc>0 THEN FOR k=1 TO nc
: GO SUB 500: PRINT AT 2,10;k:"a
Colonna": GO SUB 600: NEXT k
80 IF ns>0 THEN FOR k=1 TO ns
: GO SUB 500: PRINT AT 2,10;"Sis
tema N.":k: GO SUB 700: NEXT k
85 CLS : DIM r(13,2): DIM d$(1
3): FOR k=1 TO 13: LET d$(k)="x"
: NEXT k
90 GO SUB 1000
100 PRINT AT 18,1: INVERSE 1:"C
OMANDI": INVERSE 0:"<1> Quadro
risultati":AT 19,9:"<2> Modifica

```



```

risultati"
105 PRINT AT 20,9;"<3> Controll
o colonne";AT 21,9;"<4> Controll
o sistemi"
110 PAUSE 0: IF CODE INKEY$<49
OR CODE INKEY$>52 THEN GO TO 11
0
115 GO SUB 1000*VAL INKEY$
120 GO TO 100
500 FOR j=1 TO 13: PRINT AT j+4
j;a$(j,1, TO 3);"-";a$(j,2, TO
3);" "; NEXT j
510 RETURN
600 LET p=5: FOR l=1 TO 13
605 PAUSE 0: IF INKEY$<"1" AND
INKEY$<"2" AND INKEY$<"x" THE
N GO TO 605
610 LET c$(k,l)=INKEY$: PRINT A
T p,15;c$(k,l): LET p=p+1
620 NEXT l
625 INPUT "Confermi ? ";h$: IF
h$<"s" AND h$<"n" THEN GO TO
625
630 IF h$="n" THEN LET k=k-1:
RETURN
640 RETURN
700 LET p=5: FOR l=1 TO 13
710 INPUT s$(k,l): PRINT AT p,1
5;s$(k,l)
780 LET p=p+1: NEXT l
785 INPUT "Confermi ? ";h$: IF
h$<"s" AND h$<"n" THEN GO TO
725
790 IF h$="n" THEN LET k=k-1:
RETURN
795 RETURN
1000 CLS : PRINT "SCHEDINA TOTOC
ALCIO N.":con "Risultati parzial
i"
1010 FOR k=1 TO 13: PRINT AT k+2
j;k;AT k+2,4;a$(k,1);"-";a$(k,2
);" ";r(k,1);"-";r(k,2);" ";d$(k
); NEXT k
1100 RETURN
2000 GO SUB 1000: INPUT "num. in
contro da modificare "n: IF n<1
OR n>13 THEN GO TO 2000
2005 INPUT "risultato ";(a$(n,1)
);" ";r(n,1)
2010 INPUT "risultato ";(a$(n,2)
);" ";r(n,2)
2015 LET d$(n)=CHR$( ((r(n,1)>r(
n,2))*49)+((r(n,1)<r(n,2))*50)+((
r(n,1)=r(n,2))*120))
2020 GO SUB 1000: RETURN
3000 IF nc=0 THEN GO SUB 1000:
RETURN
3005 CLS : LET m1=0: LET cmax=1
3010 PRINT "CONTROLO COLONNE"
3020 FOR k=1 TO 13: PRINT AT 1+k
j;a$(k,1 TO 3);"-";a$(k,2, TO
3);" ";d$(k): NEXT k
3025 FOR k=1 TO nc: LET pn=0: FO
R j=1 TO 13: LET d=(d$(j)=c$(k,j
))
3030 PRINT INVERSE d;AT 1+j,k+1
0;c$(k,j): LET pn=pn+d: NEXT j
3040 LET m(k)=pn: IF pn>m1 THEN
LET m1=pn
3050 NEXT k
3055 PRINT AT 16,0;"Punti"; INVE

```

```

RSE 1;mi; INVERSE 0;" col.";
3060 FOR k=1 TO nc
3065 IF m(k)=m1 THEN PRINT k;"-
";
3070 NEXT k
3100 RETURN
4000 IF ns=0 THEN GO SUB 1000:
RETURN
4005 CLS : LET m2=0: LET smax=1
4010 PRINT "CONTROLO SISTEMI"
4020 FOR k=1 TO 13: PRINT AT 1+k
j;a$(k,1 TO 3);"-";a$(k,2, TO
3);" ";d$(k): NEXT k
4022 PRINT AT 16,4;"Punti:"
4025 FOR k=1 TO ns: LET pn=0: FO
R j=1 TO 13: LET d=0: FOR i=1 TO
3
4030 IF s$(k,j,i)=d$(j) THEN LE
T d=1: GO TO 4050
4040 NEXT i
4050 PRINT INVERSE d;AT 1+j,k*4
+7;s$(k,j): LET pn=pn+d: NEXT j
4055 PRINT AT 16,k*4+8;pn
4060 IF pn>m2 THEN LET m2=pn: L
ET smax=k
4065 NEXT k
4100 RETURN

```

Totoseikosha

2)REM linee da MERGEare
col programma "totocalcio"
di A.Russo
(DOPO averlo caricato)

```

103 DIM j$(1)
111 PRINT "8. stampa con la SE
IKOSHA 6P50-S e SAVE/LOAD delle
colonne"
120 IF j$="" OR j$<"1" OR j$>"9
" THEN GO TO 115
131 IF mn=8 THEN GO TO 2100

```

2100 REM Stampa diretta
delle schedine
con la Seikosha 6P-50S
1985 di FABIO BERNO

```

2102 REM
2104 BORDER 0: PAPER 0: INK 7: C
LS
2106 REM deve ricevere
c$(1500,13) e "col"
2108 DIM c(13): DIM r(12)
2110 DATA 175,161,146,132,117,10
3,88,74,60,45,31,16,1: RESTORE 2
110: FOR i=1 TO 13: READ c(i): N
EXT i: REM posizioni colonne
2112 DATA 143,131,118,104,92,79,
64,52,40,25,13,1: RESTORE 2112:

```



```

FOR i=1 TO 12: READ r(i): NEXT i
: REM posizioni righe
2114 LET orgX=64: LET orgY=13
2116 CLS : PRINT "1) auto-test"
: "2) stampa colonne" "3) spost
amento finestra" "4) finestra a
posizione standard" "5) salvat
aggio colonne" "6) caricamento
colonne" "0) FINE PROGRAMMA":
INPUT "scegli >"; i: IF i=1 THEN
GO SUB 2132: GO TO 2116
2118 IF i=3 THEN PRINT "vecch
io orgX : "; orgX: PRINT "vecchi
o orgY : "; orgY: INPUT "nuovo or
gX : "; orgX: INPUT "nuovo orgY :
"; orgY: GO TO 2116
2120 IF i=2 THEN GO SUB 2154: G
O TO 2116
2122 IF i=4 THEN LET orgX=64: L
ET orgY=13: PRINT "orgX=64 , o
rgY=13" "premere un tasto per c
ontinuare": PAUSE 0: GO TO 2116
2124 IF i=5 THEN CLS : DIM j$(1
0): INPUT "nome del file ? "; j$:
PRINT "SALVATO : "; j$: SAVE
j$ DATA c$( ): INPUT "Vuoi fare u
n VERIFY ? [s/n]"; k$: IF k$="s"
THEN PRINT "prepara il nast
ro e premi un ta-sto": PAUSE 0:
VERIFY j$ DATA c$( ): GO TO 2116
2126 IF i=6 THEN CLS : DIM j$(1
0): INPUT "nome del file ? "; j$:
PRINT "DA CARICARE : "; j$:
LOAD j$ DATA c$( ): GO TO 2116
2128 IF i=0 THEN STOP : REM met
tere un "GO TO 100" invece dello
"STOP" se si e' all'interno
del programma "TOTOCALCIO" di A
.RUSSO
2130 GO TO 2116
2132 REM auto-test
2134 CLS : FOR y=1 TO 12: FOR x=
1 TO 13: GO SUB 2140: NEXT x: NE
XT y
2136 INPUT "copy ? [s/n] "; i$: I
F i$="s" THEN COPY
2138 RETURN
2140 REM stampa quadrato
2142 FOR j=0 TO 6: PLOT orgX+c(x
)+j, orgY+(y): DRAW 0,7: NEXT j
2144 RETURN
2146 REM stampa sulla schedina
2148 LET j$=STR$(INT (i/4)+1):
PRINT AT 6,3: INVERSE i; j$
2150 INPUT "h: PRINT #1: FLASH 1
: premere un tasto per la stampa
h: PAUSE 0: INPUT "": COPY
2152 RETURN
2154 REM stampa colonne sistema
2156 CLS
2158 FOR i=1 TO col STEP 4
2160 INPUT "": PRINT #1: FLASH 1
: "attendi"
2162 FOR m=0 TO 3: LET j=c$(i+m
): FOR x=1 TO 13
2164 LET j=c$(i+m,x): LET y=(3*
m)+(j$="1")+2*(j$="X")+3*(j$="2"
)
2166 GO SUB 2140: NEXT x: NEXT m
2168 GO SUB 2146

```

```

2170 INPUT "": INPUT "premere "
: FLASH 1: "c": FLASH 0: " per un
'altra copia, '0' per continuare
"; i$: IF i$="c" THEN GO SUB 214
6
2171 CLS
2172 NEXT i
2174 RETURN

```

Istogrammi

```

3>REM ***** Graphics *****
** by Marco Manunta. **
5 REM
per ripartire dare GO TO 20
10 REM INIZIALIZZAZIONE
15 CLEAR 64255: LOAD "64 col."
CODE 64256
20 DIM a$(15,14): DIM a(24): L
ET zz=650
30 LET a$(1)="Premi "; LET a$(
2)="per stop": LET a$(3)="per CO
PY": LET a$(4)="omandi "; LET a
$(5)="input dati "; LET a$(6)="f
ine input": LET a$(7)="-12 "; LE
T a$(8)="-24 valori": LET a$(9)=
"barre orizz.": LET a$(10)="barr
e vertic.": LET a$(11)="rafico "
: LET a$(12)="ambia didasc.": LE
T a$(13)="0": LET a$(14)=" 50 "
: LET a$(15)="100 "
40 LET c$="Gen Feb Mar Apr Mag
Giu Lug Ago Set Ott Nov Dic "
50 FOR n=1 TO 24: LET a(n)=0:
NEXT n: LET gr=0: LET valv=12: L
ET valh=6: LET m$="100"
60 GO SUB 170: GO SUB 540
69 REM MENU'
70 IF INKEY$="i" THEN GO SUB
350
80 IF INKEY$="a" THEN LET val
v=12: LET valh=6: GO SUB 540
90 IF INKEY$="b" THEN LET val
v=24: LET valh=12: GO SUB 540
100 IF INKEY$="v" THEN LET gr=
0: GO SUB 540
110 IF INKEY$="h" THEN LET gr=
1: GO SUB 540
120 IF INKEY$="g" THEN LET gr=
2: GO SUB 540
130 IF INKEY$="c" THEN GO SUB
450
140 IF INKEY$="d" THEN COPY
150 IF INKEY$="x" THEN STOP
160 GO TO 70
169 REM PREPARAZIONE SCHERMO
170 BORDER 2: PAPER 0: INK 2: B
RIGHT 1: CLS
180 PLOT 0,143: DRAW 255,0
190 PLOT 47,143: DRAW 0,32: PLO
T 208,143: DRAW 0,32
200 PLOT 0,0: DRAW 255,0: INK 4
210 LET x=1: LET y=1: LET s$=a$
(1,1 TO 6): GO SUB zz: PRINT AT
1,4: PAPER 2: "X"

```





```

220 LET y=27: GO SUB zz: PRINT
AT 1,30: PAPER 2: "D"
230 LET x=2: LET y=1: LET s=a$
(2,1 TO 8): GO SUB zz
240 LET y=27: LET s=a$(3,1 TO
8): GO SUB zz
250 PRINT PAPER 2: AT 0,6: "C":
LET x=0: LET y=7: LET s=a$(4,1
TO 8): GO SUB zz
260 PRINT PAPER 2: AT 1,6: "I":
LET x=1: LET s=a$(5,1 TO 10): GO
SUB zz
270 PRINT PAPER 2: AT 2,6: "F":
LET x=2: LET s=a$(6,1 TO 10): GO
SUB zz
280 PRINT PAPER 2: AT 3,6: "A":
LET x=3: LET s=a$(7,1 TO 4): GO
SUB zz
290 PRINT PAPER 2: AT 3,9: "B":
LET y=10: LET s=a$(8,1 TO 10):
GO SUB zz
300 PRINT PAPER 2: AT 0,17: "H":
LET x=0: LET y=18: LET s=a$(9,
1 TO 4): GO SUB zz
310 PRINT PAPER 2: AT 1,17: "V":
LET x=1: LET s=a$(10,1 TO 4): GO
SUB zz
320 PRINT PAPER 2: AT 2,17: "8":
LET x=2: LET s=a$(11,1 TO 6):
GO SUB zz
330 PRINT PAPER 2: AT 3,17: "C":
LET x=3: LET s=a$(12,1 TO 4): GO
SUB zz
340 RETURN
349 REM INSERIMENTO DATI
350 LET x$="Scrivi il valore ma
simo,poi "+STR$(6*(valh=6 AND
gr=1)+12*(valh=12 AND gr=1)+12*(
valv=12 AND (gr=0 OR gr=2))+24*(
valv=24 AND (gr=0 OR gr=2))+" v
alori"
360 BEEP .5,30: PRINT #0: AT 1,0
;x$
370 PAUSE 80: BEEP .5,30: INPUT
"Valore max. =": LINE m$: IF LEN
m$>8 THEN GO TO 370
380 LET n=1
390 INPUT "val. n": (n); "=": LIN
E n$: IF n$="f" THEN RETURN
400 IF n$="" THEN LET n$="0"
410 LET a(n)=VAL n$
420 IF a(n)>VAL m$ THEN GO TO
390
430 IF n+1>valv AND (gr=0 OR gr
=2) THEN RETURN
431 IF n+1>valh AND gr=1 THEN
RETURN
440 LET n=n+1: GO TO 390
449 REM CAMBIAMENTO DIDASCALIE
450 LET x$="Scrivi 12 parole (m
ax. 3 car.)"
460 BEEP .5,30: PRINT #0: AT 1,0
;x$
470 PAUSE 80: BEEP .5,30: DIM h
$(1,3)
480 LET n=1: LET g=1
490 INPUT "Parola n.": (n); "=":
LINE h$(1)
495 IF h$(1)="x" THEN RETURN
500 LET c$(g TO g+3)=h$(1,1 TO
3)+" ": LET g=g+4

```

```

520 IF n+1=13 THEN RETURN
530 LET n=n+1: GO TO 490
539 REM VISUALIZZAZIONE DATI
540 FOR n=5 TO 20: PRINT AT n,1
";
NEXT n
545 IF gr=1 THEN LET x=3: LET
y=7: PRINT AT x,y; " ": LET s$="
-6 ": GO SUB zz: LET y=10: PRIN
T AT x,y; " ": LET s$="-12 ": GO
SUB zz
546 IF gr=0 OR gr=2 THEN LET x
=3: LET y=7: PRINT AT x,y; " ":
LET s$="-12 ": GO SUB zz: LET y=
10: PRINT AT x,y; " ": LET s$="-
24 ": GO SUB zz
550 IF gr=0 OR gr=2 THEN PLOT
8,16: DRAW 223,0: PLOT 23,16: DR
AW 0,119
555 IF (gr=0 OR gr=2) AND valv=
12 THEN LET x=20: LET y=4: LET
s$=c$: GO SUB zz
560 IF gr=1 THEN PLOT 8,16: DR
AW 230,0: PLOT 23,8: DRAW 0,119
570 IF gr=0 OR gr=2 THEN LET x
=19: LET y=2: LET s=a$(13,1 TO
2): GO SUB zz: LET x=12: LET y=1
: LET s=a$(14,1 TO 4): GO SUB z
z: LET x=5: LET s=a$(15,1 TO 4)
: GO SUB zz
580 IF gr=1 THEN PLOT 127,16:
DRAW 0,-8: PLOT 231,16: DRAW 0,-
8: LET x=20: LET y=2: LET s=a$(
13,1 TO 2): GO SUB zz: LET y=14:
LET s=a$(14,1 TO 4): GO SUB zz
: LET y=27: LET s=a$(15,1 TO 4)
: GO SUB zz
590 IF gr=0 OR gr=2 THEN FOR n
=47 TO 223 STEP 8/(valv/24): PLO
T n,8: DRAW 0,8: NEXT n: IF valv
=24 THEN PLOT 47-8,8: DRAW 0,8
600 IF gr=1 AND valh=12 THEN L
ET h=1: FOR g=6 TO 17: LET x=g:
LET y=1: LET s=c$(h TO h+3): GO
SUB zz: LET h=h+4: NEXT g
602 IF gr=1 AND valh=6 THEN LE
T h=1: FOR g=6 TO 17 STEP 2: LET
x=g: LET y=1: LET s=c$(h TO h+
3): GO SUB zz: LET h=h+4: NEXT g
605 IF gr=1 THEN LET y=128: FO
R n=1 TO valh: PLOT 24,y: DRAW 0
,-(6+8*(valh=6)): DRAW (a(n)/(VA
L m$/204)),0: DRAW 0,6+8*(valh=6
): DRAW -(a(n)/(VAL m$/204)),0:
LET y=y-(8+8*(valh=6)): NEXT n
610 IF gr=0 THEN LET x=32: FOR
n=1 TO valv: PLOT x,17: DRAW 6+
8*(valv=12),0: DRAW 0,(a(n)/(VAL
m$/120))-2: DRAW -(6+8*(valv=12
)),0: DRAW 0,-(a(n)/(VAL m$/120)
)+2: LET x=x+8+8*(valv=12): NEXT
n
630 IF gr=2 THEN LET x=39: PLO
T x,(a(1)/(VAL m$/114))+16: FOR
n=2 TO valv: IF a(n)>0 THEN DRA
W 8+8*(valv=12),(a(n)/(VAL m$/11
4))-(a(n-1)/(VAL m$/114)): NEXT
n
640 RETURN
649 REM ROUTINE PER LA SCRITTU=
RA A 64 COLONNE

```



```

650 LET v=y
660 FOR n=1 TO LEN s$ STEP 2
670 FOR f=0 TO 7
680 POKE USR "a"+f,16*PEEK (640
00+f+CODE s$(n)*8)+PEEK (64000+f
+CODE s$(n+1)*8): NEXT f
690 PRINT OVER 1:AT x,v;"?": L
ET v=v+1: NEXT n: RETURN
695 IF (gr=0 OR gr=1) AND valv=
12 THEN LET x=20: LET y=4: LET
s$=c$: GO SUB zz
9998 LPRINT "CODICI DA INSERIRE
A PARTIRE DALLA LOCAZIONE 642
56": "": FOR i=64256 TO (64256
+799) STEP 6: LET g$=STR$ i+"
": FOR j=0 TO 5: LET w$=STR$ (PE
EK (i+j)): FOR k=1 TO (4-LEN w$)
: LET q$=g$+"": NEXT k: LET g$=
g$+w$: NEXT j: LPRINT g$: NEXT i
: STOP
9999 CLEAR: SAVE "grafici" LINE
1: SAVE "64 col."CODE 64256,800
: VERIFY "": VERIFY "CODE

```

Codici da inserire a partire
dalla locazione 64256

```

0 0 0 0 0 0 0 0 0 2 2 2 2 0 2 0 0
0 5 5 5 5 5 5 5 0 6 6 6 6 4 7 7 8 0
0 2 7 6 7 3 7 2 0 0 D E F 2 2 4 7 8 0
0 0 8 4 F A E 0 0 2 2 2 2 0 0 0 0 0 0
0 2 4 4 4 4 4 2 0 0 0 4 2 2 2 2 2 2 0
0 0 0 6 6 6 6 6 0 0 0 0 0 2 2 2 2 2 0
0 0 0 0 0 0 0 6 6 6 0 0 0 1 2 2 2 2 4
0 6 6 B B D D D 0 0 0 2 2 2 2 2 2 7 0
0 6 9 1 2 4 F 0 0 0 0 1 2 2 6 1 9 4 0
0 2 4 6 6 9 9 6 6 0 0 0 F 1 2 2 2 4 4 0
0 6 9 6 6 9 9 6 6 0 0 0 6 6 9 9 6 6 0
0 0 0 1 2 4 2 1 0 0 0 0 0 6 6 0 2 2 4
0 0 4 2 1 2 2 4 0 0 0 2 5 1 2 0 2 2 0
0 6 9 9 9 9 9 9 0 0 0 7 9 9 9 9 9 0
0 C A E 0 0 0 0 0 0 7 9 9 9 9 9 0
0 E 9 9 9 9 9 9 0 0 0 F 0 9 9 9 9 9 0
0 F 9 9 9 9 9 9 0 0 0 7 2 2 2 2 7 7 0
0 7 2 2 2 2 4 4 0 0 9 9 A C C A A 9 0
0 4 4 4 4 4 4 4 0 0 0 9 9 F F F 9 9 0
0 9 9 9 9 9 9 9 0 0 0 6 6 0 9 9 9 9 0
0 7 2 2 2 2 2 0 0 0 0 6 6 6 6 6 6 0
0 9 9 9 9 9 9 9 0 0 0 9 9 9 9 9 9 0
0 9 9 9 9 9 9 9 0 0 0 5 5 5 5 5 5 0
0 F F 2 2 4 4 4 4 0 0 0 6 6 4 4 4 4 0
0 2 4 4 4 4 4 4 0 0 0 6 6 4 4 4 4 0
0 2 7 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0
0 4 4 4 6 5 5 5 5 3 0 0 0 2 2 4 4 3 0
0 1 1 2 7 2 2 2 0 0 0 0 3 3 5 3 1 6
0 4 4 6 5 5 5 5 2 0 0 2 2 0 5 9 9 0
0 2 2 1 1 1 1 1 0 0 0 4 5 5 6 6 0
0 2 2 2 2 2 2 2 0 0 0 9 9 F 9 9 9 0
0 F 9 9 9 9 9 9 0 0 0 0 6 9 9 9 9 0
0 7 5 5 7 4 4 0 0 0 7 5 5 7 1 1
0 3 4 4 4 4 4 0 0 0 2 4 2 1 6 0

```

```

0 2 7 2 2 2 1 0 0 0 5 5 5 7 0
0 0 5 5 5 5 5 0 0 0 0 9 9 9 6 0
0 0 5 5 5 5 5 0 0 0 0 5 5 9 2 4 0
0 0 7 1 2 4 7 0 0 0 2 4 C C C 4 2 0
0 2 2 2 2 2 2 0 0 0 4 2 3 3 2 4 0
0 0 0 5 5 A 0 0 0 0 6 6 0 0 6 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Joystick

1 REM ROUTINE IN L.M.
PER L'USO DEI JOYSTICK
TIPO KEMPSTON

di Raffaele Ilardo

(le coordinate linea,colonna
sono a 23550 e 23551 rispettivam
ente; il flag "fuoco" e'a 23449
)

```

2 DATA 219,31,254,0,200,245,1
75,50,253,91,241,254,16,56,5,50,
253,91,214,16,42,254,91,254,1,32
,1,36,254,2,32,1,37
3 DATA 254,4,32,1,44,254,8,32
,1,45,254,5,32,2,36,44,254,10,32
,2,37,45,254,6,32,2,44,37,254,9,
32,2,45,36,125,254,22,32,1
4 DATA 45,254,255,32,1,44,124
,254,32,32,1,37,254,255,32,1,36,
34,254,91,201
5 FOR n=23456 TO 23548: READ
d: POKE n,d: NEXT n
6 STOP
10 REM segue breve programma
dimostrativo (n.1)
11 PRINT AT PEEK 23550,PEEK 23
551;"X"
12 IF PEEK 23549>0 THEN PRINT
AT PEEK 23550,PEEK 23551;CHR$ 1
28
13 RANDOMIZE USR 23456: GO TO
11
20 REM programma esempio n.2
21 BORDER 6: CLS : PLOT 83,7:
DRAW 172,0: DRAW 0,-7: DRAW INK
1:-172,0: DRAW INK 1;0,7: LET
p=0: POKE 23550,12: POKE 23551,1
6
22 LET l=PEEK 23550: LET c=PEE
K 23551: PRINT AT l,c: INK 1;"Y"
23 RANDOMIZE USR 23456: IF (PE
EK 23550<)>1 OR PEEK 23551<>c) AN
D ATTR (PEEK 23550,PEEK 23551)<>
57 THEN LET p=p+1: PRINT AT 21,
0: INK 1;"PUNTI:";p: GO TO 22
24 IF PEEK 23550=1 AND PEEK 23
551=c THEN GO TO 23
25 PRINT AT 21,0: FLASH 1;"PUN
TI:";p: PRINT #0:"press any key
to restart": BEEP .5,30: PAUSE
0: CLS : GO TO 21

```



```

A9C0 DB1F      IN  A,(#1F)
A9C2 FE00     CP   #00
A9C4 C8       RET  Z
A9C5 F5       PUSH AF
A9C6 AF       XOR  A
A9C7 32FD5B   LD   (#5BFD),A
A9CA F1       POP  AF
A9CB FE10     CP   #10
A9CD 3805     JR   C,LA9D4
A9CF 32FD5B   LD   (#5BFD),A
A9D2 D610     SUB  #10
A9D4 2AFE5B   LA9D4 LD HL,(#5BFE)
A9D7 FE01     CP   #01
A9D9 2001     JR   NZ,LA9DC
A9DB 24       INC  H
A9DC FE02     LA9DC CP #02
A9DE 2001     JR   NZ,LA9E1
A9E0 25       DEC  H
A9E1 FE04     LA9E1 CP #04
A9E3 2001     JR   NZ,LA9E6
A9E5 2C       INC  L
A9E6 FE08     LA9E6 CP #08
A9EB 2001     JR   NZ,LA9EB
A9EA 2D       DEC  L
A9EB FE05     LA9EB CP #05
A9ED 2002     JR   NZ,LA9F1
A9EF 24       INC  H
A9F0 2C       INC  L
A9F1 FE0A     LA9F1 CP #0A
A9F3 2002     JR   NZ,LA9F7
A9F5 25       DEC  H
A9F6 2D       DEC  L
A9F7 FE06     LA9F7 CP #06
A9F9 2002     JR   NZ,LA9FD
A9FB 2C       INC  L
A9FC 25       DEC  H
A9FD FE09     LA9FD CP #09
A9FF 2002     JR   NZ,LAA03
AA01 2D       DEC  L
AA02 24       INC  H
AA03 7D       LAA03 LD A,L
AA04 FE16     CP   #16
AA06 2001     JR   NZ,LAA09
AA08 2D       DEC  L
AA09 FEFF     LAA09 CP #FF
AA0B 2001     JR   NZ,LAA0E
AA0D 2C       INC  L
AA0E 7C       LAA0E LD A,H
AA0F FE20     CP   #20
AA11 2001     JR   NZ,LAA14
AA13 25       DEC  H
AA14 FEFF     LAA14 CP #FF
AA16 2001     JR   NZ,LAA19
AA18 24       INC  H
AA19 22FE5B   LAA19 LD HL,#5BFE,HL
AA1C C9       RET

```

Music comp.

```

916 PRINT AT 15,0;"6) GRAPHIC M
ELDY"
986 IF s#="6" THEN GO TO 5500
5500 CLS :
5501 IF z(h)-1>=61 AND z(h)-1<=6
6 THEN LET ot=3: GO TO 5508

```

```

5502 IF z(h)-1=67 AND z(h)-1<=1
00 THEN LET ot=2: GO TO 5508
5503 IF z(h)-1>=50 AND z(h)-1<=6
0 THEN LET ot=4: GO TO 5508
5504 LET ot=5
5508 RESTORE 6100
5509 GO SUB 6100
5510 GO SUB 6010
5520 CLS
5540 PLOT 0,8: DRAW 255,0
5550 PLOT 0,8: DRAW 0,167
5560 LET rip=0: LET s=z(1)
5570 LET v=0
5580 DIM h(s-1)
5590 FOR l=1 TO s-1
5600 IF d#="1" THEN GO SUB 5970
: LET f=0: LET ju=50
5610 IF d#="2" THEN GO SUB 5990
: LET f=3: LET ju=20
5630 NEXT l
5640 IF d#="2" THEN GO TO 5810
5650 LET e=-180
5660 FOR p=1 TO s-1
5670 IF e>h(p) THEN GO TO 5690
5680 LET e=h(p)
5690 NEXT p
5700 LET g=240
5710 FOR p=1 TO s-1
5720 IF g<h(p) THEN GO TO 5740
5730 LET g=h(p)
5740 NEXT p
5750 PRINT AT 0,0;"min.";g/5
5760 PRINT AT 0,22;"max.";e/5
5765 PRINT AT 0,12;CHR# 144;".";
(e/5)-(g/5)
5770 PLOT 0,g+ju: DRAW 255,0
5780 FOR k=g+ju TO e+ju STEP 5
5790 PLOT 0,k: DRAW 255,0
5800 NEXT k
5810 FOR p=2 TO s-1
5820 IF h(p)<h(p-1) THEN LET i=
h(p-1)-h(p): LET i=-i: GO TO 585
0
5830 IF h(p)>h(p-1) THEN LET i=
h(p)-h(p-1): GO TO 5850
5840 IF h(p)=h(p-1) THEN LET i=
0
5850 LET v=v+ot
5860 CIRCLE v,h(p-1)+ju,1
5870 PLOT v,h(p-1)+ju: DRAW ot,i
5880 NEXT p
5890 IF rip=1 THEN GO TO 5920
5900 PRINT #0;"PREMI 1 o 2 o E o
S"
5920 PAUSE 0
5940 IF INKEY#="1" THEN LET d#="
1": LET rip=1: GO TO 5570
5950 IF INKEY#="2" THEN LET d#="
2": LET rip=1: GO TO 5570
5955 IF INKEY#="E" THEN LET et=
1: GO TO 2270
5956 IF INKEY#="S" THEN COPY
5960 GO TO 9
5970 LET h(1)=n(1,o)*5
5980 RETURN
5990 LET h(1)=VAL p$(1,o)*64
6000 RETURN
6010 PRINT BRIGHT 1;AT 0,8;"GRA
PHIC MELODY"
6020 PRINT AT 5,3;"1) GRAFICO LI
NEA MELODICA"

```

```
6030 PRINT AT 8,3;"2) GRAFICO LI
NEA RITMICA"
6040 PAUSE 0
6050 LET d$=INKEY$
6060 IF d$="1" OR d$="2" THEN R
RETURN
6070 GO TO 6040
6080 LET h(1)=(n(1,o)*5)+(VAL p$
(1,o)*64)
6090 RETURN
6100 DATA 0,103,73,73,62,8,8,8
6110 FOR p=0 TO 7: READ t: POKE
USR "a"+p,t: NEXT p
6120 RETURN
```

```
1015 FOR C=1 TO 10
1020 LET D=(S*10)+C: GO SUB 8000
: INPUT #10;D$
1030 GO SUB (20+C): PRINT Z$;D$
1040 NEXT C
1060 INPUT "VUOI CONTINUARE ? s/
n";I$: IF I$="s" THEN GO TO 101
0
1070 STOP
7998 REM #####
7999 REM SUBROUTINE DI MODIFICA
VARIABILI DI SISTEMA PER LA RICE
RCA RANDOM
8000 LET PR=P(D): LET PA=PR-512*
(INT (PR/512)): LET R=INT (PR/51
2)
8005 LET k=PEEK 23857: POKE 2391
1,0
8010 IF k<>R THEN POKE 23855,0:
POKE 23856,2: POKE 23857,R-1: I
NPUT #10;S$
8020 POKE 23855,(PA-256*INT (PA/
256)): POKE 23856,INT (PA/256):
RETURN
8050 REM #####
8051 REM SUBROUTINE DI PRINT NEL
FILE
8060 LET LD=(LEN D$)+1: LET IRD=
IRD+LD: LET DIN=DIN+1
8070 LET P(DIN)=IRD: PRINT #10;D
$: RETURN
```

Microdrives

```
5 REM listato n.1
10 REM DEFINIZIONE DELLE VARIA
BILI E DELLE STRINGHE BASE
11 REM DIM P(1000) := MATRICE
CHE TIENE LA POSIZIONE RELATIVA
DEL DATO
12 REM DIN :=DATI INSERITI
13 REM IRD := INDIRIZZO
RELATIVO DEL DATO CORRENTE
21 LET Z$="NOME " :
RETURN
22 LET Z$="COGNOME " :
RETURN
23 LET Z$="TELEFONO " :
RETURN
24 LET Z$="CITTA' " :
RETURN
25 LET Z$="FRAZIONE " :
RETURN
26 LET Z$="DATA DI NASCITA " :
RETURN
27 LET Z$="PROFESSIONE " :
RETURN
28 LET Z$="INTERESSI " :
RETURN
29 LET Z$="NOTE VARIE " :
RETURN
30 LET Z$="VARIE " :
RETURN
35 CLEAR #: DIM P(1000): LET D
IN=1: LET IRD=0: LET P(1000)=IRD
40 OPEN #10:"M":1:"IND.1"
41 REM #####
50 REM ROUTINE DI INSERIMENTO
DATI
60 FOR C=1 TO 10
70 GO SUB (20+C): INPUT (Z$),D
$: PRINT Z$;D$: GO SUB 8050
80 NEXT C
150 INPUT "VUOI INSERIRE ANCORA
? s/n";I$: IF I$="s" THEN GO T
O 60
160 CLOSE #10: STOP
999 REM #####
1000 REM ROUTINE DI STAMPA DATI
1005 OPEN #10:"M":1:"IND.1"
1010 INPUT "QUALE RECORD STAMPO
? " :S
1012 CLS
```

```
5 REM LISTATO N.2
10 REM DEFINIZIONE DELLE VARIA
BILI E DELLE STRINGHE BASE
12 REM DIN :=DATI INSERITI
20 LET A$="
": GO TO 40
21 LET Z$="NOME " :
RETURN
22 LET Z$="COGNOME " :
RETURN
23 LET Z$="TELEFONO " :
RETURN
24 LET Z$="CITTA' " :
RETURN
25 LET Z$="FRAZIONE " :
RETURN
26 LET Z$="DATA DI NASCITA " :
RETURN
27 LET Z$="PROFESSIONE " :
RETURN
28 LET Z$="INTERESSI " :
RETURN
29 LET Z$="NOTE VARIE " :
RETURN
30 LET Z$="VARIE " :
RETURN
40 CLEAR #: OPEN #10:"M":1:"IN
D.1"
41 REM #####
50 REM ROUTINE DI INSERIMENTO
DATI
60 FOR C=1 TO 10
70 GO SUB (20+C): INPUT (Z$),D
$
72 LET D$=D$+A$(1 TO (20-LEN D
$))
```



listabi

```

4100 INPUT "QUALE ? ";W
4110 IF W=1 THEN GO SUB 40
4120 IF W=2 THEN GO SUB 2000
4130 IF W=3 THEN GO SUB 4800
4140 IF W=4 THEN STOP
4200 GO TO 4010
4210 STOP
4800 REM MODIFICA PARAMETRI
4810 CLS : INPUT "LUNGHEZZA CAMP
O ? ";LC
4820 CLS : INPUT "NUMERO CAMPI ?
";NC
4830 CLS : INPUT "NUMERO RECORDS
? ";NR
4835 GO SUB 9000
4840 RETURN
4999 STOP
7998 REM *****
7999 REM SUBROUTINE DI MODIFICA
VARIABILI DI SISTEMA PER LA RIC
ERCA RANDOM
8000 LET PR=D*(LC+1); LET PA=PR-
512*(INT (PR/512)); LET R=INT (P
R/512)
8005 LET k=PEEK 23857; POKE 2391
1,0
8010 IF k<>R THEN POKE 23855,0;
POKE 23856,2; POKE 23857,R-1; I
NPUT #10;R$
8020 POKE 23855,(PA-256*INT (PA/
256)); POKE 23856,INT (PA/256);
RETURN
9000 LET S$=""; LET A$=""; FOR J
=1 TO LC: LET A$=A$+" "; NEXT J:
RETURN

```

Checksum

```

5 REM CHECKload (C) Mellina
10 CLEAR 31999
20 FOR n=32000 TO 32151: READ
x: POKE n,x: NEXT n
30 DATA 24,26,108,117,110,103,
104,46,112,114,111,103,114,46,32
6,108,105,110,101,32,108,101,11
0,32,99,104,107
40 DATA 197,62,2,205,1,22,225,
35,35,6,13
50 DATA 126,215,35,16,251,229,
229,42,75,92,237,75,83,92,175,23
7,66,30,32,1,240,216,205,42,25,1
25,50,177,92,124,33,176,92,119,2
05,40,26,14,2,225,6,13
60 DATA 126,215,35,16,251,13,3
2,245,42,83,92
70 DATA 229,237,75,75,92,175,2
37,66,225,200,62,6,215,205,40,26
35,126,50,177,92,35,126,229,33,
176,92,119,205,40,26,225,237,75,
176,92,175
80 DATA 35,142,16,252,13,40,24
9,229,50,177,92,33,176,92,54,0,2
05,40,26,225,35,24,196
100 PRINT AT 2,0;"SAVE ""CHECKC
"" CODE": SAVE "CHECKC"CODE 3200
0,152

```

```

64000 181A
64002 6C756E
64005 67682E
64008 70726F
64011 67722E
64014 20066C
64017 696E55
64020 286C65
64023 6E2063
64026 686B
64028 C5
64029 3E02
64031 C00116
64034 E1
64035 23
64036 23
64037 060D
64039 7E
64040 D7
64041 23
64042 10FB
64044 E5
64045 E5
64046 2A4B5C
64049 ED4B535C
64053 AF
64054 ED42
64056 1E20
64058 01F008
64061 CD2A19
64064 7D
64065 32B15C
64068 7C
64069 21B05C
64072 77
64073 CD281A
64076 0E02
64078 E1
64079 060D
64081 7E
64082 D7
64083 23
64084 10FB
64086 0D
64087 20F5
64089 2A535C
64092 E5
64093 ED4B4B5C
64097 AF
64098 ED42
64100 E1
64101 C8
64102 3E06
64104 D7
64105 CD281A
64108 23
64109 7E
64110 32B15C
64113 23
64114 7E
64115 E5
64116 21B05C
64119 77
64120 CD281A
64123 E1
64124 ED4B05C
64128 AF
64129 23
64130 8E
64131 10FC
64133 0D
JR LFA1C
DEFB "1","u","n"
DEFB "g","h","o"
DEFB "p","r","o"
DEFB "q","r","o"
DEFB "i","#06","i"
DEFB "i","n","e"
DEFB "1","1","e"
DEFB "h","1","c"
DEFB "h","k"
LFA1C PUSH BC
LD A,#02
CALL 05633
POP HL
INC HL
INC HL
INC HL
LD B,#0D
LFA27 LD A,(HL)
RST #10
INC HL
DJNZ LFA27
PUSH HL
PUSH HL
LD HL,(23627)
LD BC,(23635)
XOR A
SBC HL,BC
LD E,#20
LD BC,55536
CALL 06442
LD A,L
LD (23729),A
LD A,H
LD HL,23728
LD (HL),A
CALL 06696
LD C,#02
LFA4E POP HL
LD B,#0D
LFA51 LD A,(HL)
RST #10
INC HL
DJNZ LFA51
DEC C
JR NZ,LFA4E
LD HL,(23635)
LFA5C PUSH HL
LD BC,(23627)
XOR A
SBC HL,BC
POP HL
RET Z
LD A,#06
RST #10
CALL 06696
INC HL
LD A,(HL)
LD (23729),A
LD A,(HL)
POP HL
LD BC,(23728)
CALL 06696
POP HL
LD A,(HL)
XOR A
LFA81 INC A,(HL)
DJNZ LFA81
DEC C

```



```

64134 28F9
64136 E5
64137 32B15C
64140 21805C
64143 3600
64145 CD281A
64148 E1
64149 23
64150 18C4
64152 00

```

```

JR Z,LFA81
PUSH HL
LD (23729),A
LD HL,23728
LD (HL),#00
CALL 06696
POP HL
INC HL
JR LFA5C
NOP

```

Un completo disassemblatore

1 RUN 1000: REM

DISASSEMBLATORE

(C)G.M.Mellina

Auto-Data

```

9000 LET F=44000: LET T=44151: F
OR n=F TO T: READ x: POKE n,x: N
EXT n
9001 DATA 000,000,000,000,000,00
0,000,000
9002 DATA 000,000,000,000,000,00
0,000,000
9003 DATA 000,000,000,000,000,00
0,000,000
9004 DATA 000,000,000,000,000,00
0,000,000
9005 DATA 000,000,000,000,000,000
0,000,000
9006 DATA 000,000,000,000,000,000
0,000,000
9007

```

```

10 INPUT "ENTER or STOP ", LIN
E z$: IF z$=" " STOP " THEN STOP
: GO TO 10
15 POKE 23658,8: INPUT "esadec
.? (H)=>",x$,"DEFB? (B)=>",y$
20 CLS : INPUT AT 0,0;"indir.s
tart =>",i,"indir.end =>",e
25 IF y$="B" THEN GO TO 705
30 LET x=PEEK i: LET y=PEEK (i
+1): IF x=237 THEN GO TO 200
40 IF x=203 THEN GO TO 300
50 IF x=221 OR x=253 THEN GO
TO 400
100 REM norm
110 LET l=1: IF x>63 AND x<192
AND x<>118 THEN GO TO 150
120 LET n=1: LET y=62: GO SUB 9
00: FOR y=4 TO 9: IF i$(y)=" " TH
EN GO TO o
131 IF i$(y)="#" THEN LET s$=s
$+STR$ PEEK (i+1): LET l=2: GO T
O 135
132 IF i$(y)="@" THEN LET s$=s
$+STR$(PEEK (i+1)+256*PEEK (i+2
)): LET l=3: GO TO 135
133 IF i$(y)="*" THEN LET l=2:
GO TO 140
134 LET s$=s$+i$(y)
135 NEXT y: IF z$="E3" THEN LE
T s$=s$+"L"
136 GO TO o
140 IF PEEK (i+1)>127 THEN LET
s$=s$+STR$(PEEK (i+1)-256)+"=">
"+STR$(i+d-254+PEEK (i+1)): GO
TO o
141 LET s$=s$+" "+STR$ PEEK (i+
1)+"=">"+STR$(i+2+d+PEEK (i+1)):
GO TO o
150 IF x<128 THEN LET z1=1: GO
TO 160
151 LET z1=INT (x/8)-14
160 LET s$=o$(CODE a$(z1)-48)+"

```

```

...00,000,000,000,000,00
...000,000,000,000,000,000
DATA 000,000,000,000,000,000,00
9018 DATA 000,000,000,000,000,000,00
0,000,000
9019 DATA 000,000,000,000,000,000,00
0,000,000
9100 REM inizio progr. unload
9110 CLS : PRINT AT 8,2;"STOP TH
E TAPE and";AT 10,5;"press any
key to RUN": PAUSE 0: PRINT ".
.wait ..."
9120 LET i=PEEK 23635+256*PEEK 2
3636: LET i=i+PEEK (i+2)+4
9130 FOR x=0 TO 18: LET i=i+5: F
OR y=0 TO 7
9140 LET n=PEEK (64000+y+8*x): L
ET n$=STR$ n
9150 IF LEN n$<3 THEN LET n$="0
"+n$: GO TO 9150
9160 POKE i,CODE n$(1): POKE i+1
,CODE n$(2): POKE i+2,CODE n$(3)
: POKE i+6,n
9170 LET i=i+10: NEXT y: NEXT x
9180 POKE 23681,INT (1/256)
9190 POKE 23627,i-256*PEEK 23681
: POKE 23628,PEEK 23681
9200 RUN 9999

```

```

": IF z1=1 THEN LET z2=INT (x/
B)-7: GO TO 180
170 IF z1=2 OR z1=3 OR z1=5 THE
N LET z2=8: GO TO 180
175 GO TO 185
180 LET s$=s$+b$(z2): IF b$(z2)
="(" THEN LET s$=s$+"HL)"
181 LET s$=s$+" "
185 LET z2=x-8*INT (x/8)+1: LET
s$=s$+b$(z2): IF b$(z2)="(" THE
N LET s$=s$+"HL)"
190 GO TO o
200 REM ED
210 LET x=y: LET l=2: LET n=126
: LET y=29: GO SUB 900: FOR y=4
TO 9: IF i$(y)=" " THEN GO TO o
260 IF i$(y)="@" THEN LET s$=s
$+STR$(PEEK (i+2)+256*PEEK (i+3
)): LET l=4: GO TO 280
270 LET s$=s$+i$(y)
280 NEXT y: GO TO o
300 REM CB
310 LET x=y: IF x<48 THEN LET
z1=INT (x/8)+1: GO TO 340
321 IF x<56 THEN GO TO 700
322 IF x<64 THEN LET z1=7: GO
TO 340
330 LET z1=INT (x/64)+7
340 LET z2=x-8*INT (x/8)+1: LET
z3=INT (x/8)-8*INT (x/64): LET
s$=o$(CODE c$(z1)-48)+" ": IF x>
63 THEN LET s$=s$+STR$ z3+" "
370 LET s$=s$+b$(z2): IF b$(z2)
="(" THEN LET s$=s$+"HL)"
380 LET l=2: GO TO o
400 REM DD/FD
410 IF y=203 THEN GO TO 500
420 LET x=y: LET l=2: LET n=185
: LET y=19: GO SUB 900: FOR y=4
TO 9: IF i$(y)=" " THEN GO TO o
450 FOR y=4 TO 9: IF i$(y)=" " T
HEN GO TO o
460 IF i$(y)="@" THEN LET s$=s
$+STR$(PEEK (i+3): LET l=4: GO T
O 480
461 IF i$(y)="@" THEN LET s$=s
$+STR$(PEEK (i+2)+256*PEEK (i+3
)): LET l=4: GO TO 480
462 IF i$(y)="*" THEN LET l=3:
GO TO 490
470 LET s$=s$+i$(y): IF i$(y)<>
"I" THEN GO TO 480
472 IF PEEK i=221 THEN LET s$=
s$+"X": GO TO 480
473 LET s$=s$+"Y"
480 NEXT y: GO TO o
490 IF PEEK (i+2)>127 THEN LET
s$=s$+"-": GO TO 492
491 LET s$=s$+"+"
492 LET s$=s$+STR$(PEEK (i+2)):
GO TO 480
500 REM DDCB/FDCB
510 LET x=PEEK (i+3): LET l=4:
GO SUB 600: IF NOT (z$(2)="6" OR
z$(2)="E") OR z$="36" THEN GO
TO 700
520 IF x<48 THEN LET z1=INT (x
/8)+1: GO TO 540
530 LET z1=INT (x/64)+7: IF x=6
2 THEN LET z1=7
540 LET z2=INT (x/8)-8*INT (x/6
4): LET s$=o$(CODE c$(z1)-48)+"
": IF PEEK i=221 THEN LET z$="X
": GO TO 570
550 LET z$="Y"
570 IF x>63 THEN LET s$=s$+STR
$ z2+" "
580 LET s$=s$+"(i+z$: IF PEEK
(i+2)>127 THEN LET s$=s$+"-": G
O TO 590
581 LET s$=s$+"+"
590 LET s$=s$+STR$(PEEK (i+2)+"
"): GO TO o
600 REM dec-hex
610 LET z1=INT (x/16): LET z2=x
-16*z1: LET z$=h$(z1+1)+h$(z2+1)
: RETURN
700 REM DEFb
705 GO SUB 710: GO TO 810
710 IF PEEK i>32 THEN LET s$="
">+CHR$(PEEK i): GO TO 712
711 LET s$=STR$(PEEK i)
712 LET s$="DEFb "+s$: LET l=1:
GO SUB 850: RETURN
800 REM end
805 GO SUB 850
810 LET i=i+1: IF i>e THEN GO
TO 10
815 IF y$="B" THEN GO TO 705
820 IF PEEK (i-1)=207 THEN GO
TO 700
830 IF PEEK (i-1)<>239 THEN GO
TO 30
835 IF PEEK i=56 THEN GO TO 70
0
840 GO SUB 710: LET i=i+1: GO T
O 835
850 IF PEEK 23560=82 THEN GO T
O 20
860 PRINT i+d;TAB 6:; FOR y=0 T
O l-1: LET x=PEEK (i+y): IF x$<>
"H" THEN PRINT x;" "; GO TO 87
0
865 GO SUB 600: PRINT z$:
870 NEXT y: PRINT TAB 15;s$:TAB
0;#;AT 0,10;"press "R" per r

```



STAMPANTI Epson,

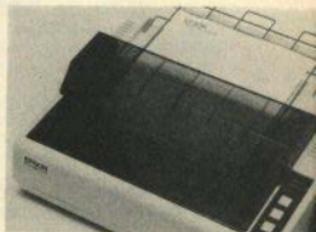
una scelta prestigiosa, senza compromessi

Scegliere una stampante è facile?

C'è una sola regola, pretendere sempre il massimo delle prestazioni, senza compromessi: materiali e componenti di prima qualità, disegno elegante, grande affidabilità, robustezza, facilità e flessibilità d'impiego, prezzo adeguato e la garanzia di un grande costruttore leader mondiale.

Così, con Epson, la scelta è facile e sicura.

Epson il più grande costruttore al mondo di stampanti vi offre una gamma di prodotti prestigiosi che soddisfano ogni vostra necessità. Epson, una soluzione raffinata, in esclusiva per il vostro computer.



FX-80

Indispensabile nelle applicazioni in cui la versatilità e la qualità di stampa sono un imperativo.

Possibilità di creare qualsiasi carattere su una matrice di 11x9 punti. Memoria RAM da 4 Kbyte. 256 tipi di caratteri definibili dall'utente. 136 tipi di caratteri a corredo. Alta velocità di stampa a 160 caratteri al secondo su 80 colonne. Ben 9 modi di stampa grafica punto a punto selezionabili sulla stessa riga contemporaneamente.

Inseritori automatici di fogli singoli a singola e doppia vaschetta.

FX-100

Con 132 colonne e 160 caratteri al secondo, la FX-100 è la stampante ideale per data processing e tabelle, specialmente in ufficio grazie anche alla possibilità di accettare fogli singoli e moduli continui di qualsiasi formato. La matrice di punti 11x9 consente prestazioni grafiche e la formazione di una grandissima varietà di tipi di carattere, fino a 256, definibili anche dall'utente e memorizzati nei 3 kbyte di RAM interna. La FX-100 non teme la fatica: la testina di stampa è garantita per oltre 100 milioni di caratteri ed è facilmente sostituibile. Inseritori automatici di fogli singoli.

AX-80 / AX-100

Le migliori prestazioni da stampanti, semplici, versatili, silenziose e veloci con 100 caratteri al secondo. 128 tipi di caratteri selezionabili e 11 set internazionali. 80 o 132 colonne. 6 diverse possibilità grafiche. Tutti i tipi di carta, modulo continuo, foglio singolo. Inseritori automatici di fogli singoli.



Epson dunque, senza compromessi.

EPSON

EPSON CORPORATION
HEAD OFFICE
8-6, Honcho, Shinjy-ku, Nagano
399-07 JAPAN



EPSON, computer e periferiche
sono prodotti distribuiti,
assistiti e garantiti
da SECI S.p.A. - Milano - Via Timavo, 12

segi SERVIZI
GENERALI PER
L'INFORMATICA

```

estart";: RETURN
900 GO SUB 600: LET x=n+y: GO S
UB 950: LET x=n+INT ((y+1)/2): G
O SUB 950: FOR x=n TO n+y/2: IF
z#t$(x,1 TO 2) THEN LET i#t$(
x): LET s#o$(CODE i$(3)-48)+"
": RETURN
910 NEXT x: IF x>126 THEN GO T
O 700
920 STOP
950 IF z#>t$(x,1 TO 2) THEN L
ET n=x
960 RETURN
1000 BORDER 5: CLEAR 34099: LET
h#="0123456789ABCDEF": LET a#="N
21r13sU7": LET b#="BCDEHL(A)": LE
T c#="chafopq4ln": LET o=800: DI
M t$(223,9): FOR x=1 TO 223: REA
D t$(x): NEXT x: DIM o$(67,4): F
OR x=1 TO 67: READ o$(x): NEXT x
: REM d = +/- displacement
1001 LET d=0: GO TO 10
1020 REM
1030 DATA "00T", "01NBC,@", "02N(B
C),A", "03GBC", "04GB", "05>B", "06N
B,#", "07d", "08BAF,AF'", "092HL,BC
", "0ANA,(BC)", "0B>BC", "0CGC", "0D
>C", "0ENC,#", "0F1"
1031 DATA "10e*", "11NDE,@", "12N
(DE),A", "13GDE", "14GD", "15>D", "1
6ND,#", "17b", "18M*", "192HL,DE", "
1ANA,(DE)", "1B>DE", "1CGE", "1D>E"
", "1ENE,#", "1Fq"
1032 DATA "20MNZ,*", "21NHL,@", "2
2N(@),HL", "23GHL", "24GH", "25>H",
"26NH,#", "27=", "28MZ,*", "292HL,H
L", "2ANHL,(@)", "2B>HL", "2CGL", "2
D>L", "2ENL,#", "2F<"
1033 DATA "30MNC,*", "31NSP,@", "3
2N(@),A", "33GSP", "34G(HL)", "35>(
HL)", "36N(HL),#", "37m", "38MC,#",
"392HL,SP", "3ANA,(@)", "3B>SP", "3
CBA", "3D>A", "3ENA,#", "3F6"
1034 DATA "76D"
1035 DATA "C0^NZ", "C1[BC", "C2LNZ
,@", "C3L@", "C45NZ,e", "C5\BC", "C6
2A,#", "C7k0", "C8^Z", "C9^", "CALZ,
e", "CC5Z,e", "CD5e", "CE1A,#", "CFk
8"
1036 DATA "D0^NC", "D1[DE", "D2LNC
,@", "D3X(#),A", "D45NC,@", "D5\DE"
", "D6r#", "D7k16", "D8^C", "D9C", "DA
LC,@", "DBFA,(#)", "DC5C,e", "DE1A,
#", "DFk24"
1037 DATA "E0^PO", "E1[HL", "E2LPO
,@", "E3B(SP),HL", "E45PO,@", "E5\H
L", "E63#", "E7k32", "E8^PE", "E9L(H
L)", "EALPE,@", "EBBDE,HL", "EC5PE,

```

```

@", "EEs#", "EFk40"
1038 DATA "F0^P", "F1[AF", "F2LP,@
", "F3?", "F45P,@", "F5\AF", "F6U#",
"F7k48", "F8^M", "F9NSP,HL", "FALM,
@", "FBA", "FC5M,@", "FE7#", "FFk56"
2000 REM after ED
2020 DATA "40FB,(C)", "41X(C),B",
"421HL,BC", "43N(@),BC", "44S", "45
", "46E0", "47NI,A", "48FC,(C)", "4
9X(C),C", "4A1HL,BC", "4BNBC,(@)",
"4D", "4FNR,A"
2021 DATA "50FD,(C)", "51X(C),D",
"521HL,DE", "53N(@),DE", "56E1", "5
7NA,I", "58FE,(C)", "59X(C),E", "5A
1HL,DE", "5BNDE,(@)", "5EE2", "5FNA
,R"
2022 DATA "60FH,(C)", "61X(C),H",
"621HL,HL", "63N(@),HL", "67j", "68
FL,(C)", "69X(C),L", "6A1HL,HL", "6
BNHL,(@)", "6Fe"
2023 DATA "70FF,(C)", "721HL,SP",
"73N(@),SP", "78FA,(C)", "79X(C),A
", "7A1HL,SP", "7BNSP,(@)"
2024 DATA "A0Q", "A1:", "A2J", "A3
Z", "A80", "A98", "AAH", "ABY"
2025 DATA "B0R", "B1:", "B2K", "B3W
", "BBP", "B99", "BAI", "BBV"
3000 REM after DD/FD
3020 DATA "0921,BC", "1921,DE", "2
1NI,@", "22N(@),I", "23GI", "292I,I
", "2ANI,(@)", "2B>I", "34G(I*)", "3
5>(I*)", "36N(I*)", "392I,SP"
3021 DATA "46NB,(I*)", "4ENC,(I*)
", "56ND,(I*)", "5ENE,(I*)", "66NH,
(I*)", "6ENL,(I*)", "70N(I*)", "B", "7
1N(I*),C", "72N(I*)", "D", "73N(I*)",E
", "74N(I*),H", "75N(I*),L", "77N(I
*)", "A", "7ENA,(I*)"
3022 DATA "862A,(I*)", "8E1A,(I*)
", "96r(I*)", "9E1A,(I*)", "A63(I*)
", "AeS(I*)", "86U(I*)", "BE7(I*)",
"E1I", "E3B(SP),I", "E5\I", "E9L(I
)", "F9NSP,I"
4000 REM cod.op.
4010 DATA "ADC", "ADD", "AND", "BIT
", "CALL", "CCF", "CP", "CPD", "CPDR",
"CPF", "CFIR", "CPL", "DAA", "DEC",
"DI", "DJNZ", "EI", "EX", "EXX", "HAL
T", "IM", "IN", "INC", "IND", "INDR",
"INI", "INIR", "JP", "JR", "+CHR# B",
"LD", "LDD", "LDDR", "LDI", "LDIR"
4011 DATA "NEG", "NOP", "OR", "OTDR
", "OTIR", "OUT", "OUTD", "OUTI", "PO
P", "PUSH", "RES", "RET", "RETI", "RE
TN", "RL", "RLA", "RLC", "RLCA", "RLD
", "RR", "RRA", "RRC", "RCCA", "RRD",
"RST", "SBC", "SCF", "SET", "SLA", "S
RA", "SRL", "SUB", "XOR"

```



Le variabili di sistema della ROM-ombra

Quando si chiama un comando del basic esteso, nella mappa di memoria dello Spectrum avvengono alcune variazioni. La principale è l'aggiunta di una serie di nuove variabili di sistema in coda a quelle normalmente presenti, dalla locazione 23552 alla locazione 23733. Con queste aggiunte le variabili di sistema vengono a occupare ulteriori 58 bytes, fino alla locazione 23791.

23734. Questa locazione contiene una serie di flag, cioè di indicatori logici. Ogni singolo bit della variabile segnala che è in corso una determinata operazione che coinvolge la ROM-ombra installata nell'interfaccia 1. In particolare, i bit più significativi (7,6,5,4) controllano le operazioni di microdrive (sono posti a 1 rispettivamente durante VERIFY, MERGE, SAVE, LOAD); il bit 3 è acceso quando è in funzione la rete locale, il bit 2 quando è alterato ERRSP (vedi le normali variabili di sistema), il bit 1 quando si esegue un CLEAR, il bit 0, infine, quando si esegue un qualsiasi comando del basic esteso.

23735. Questa variabile occupa due bytes, ed è una delle più interessanti per il programmatore in linguaggio macchina. Essa permette infatti di estendere ulteriormente il basic dello Spectrum.

Normalmente la variabile contiene l'indirizzo della routine di gestione degli errori. Facendola puntare alle vostre routine, potrete far passare comandi di solito non ritenuti validi. Comunque le cose non sono proprio così semplici, e l'argomento è molto vasto, quindi ne tratteremo in altra sede (cfr., per un esempio, «Absolute DRAW», di G. Carri, su S.C. n. 07).

23737. Più che una variabile, questa è una vera e propria subroutine, che occupa una decina di bytes. Serve alla ROM-ombra per poter chiamare le routine della normale ROM 16K.

23747. Quando si usa la RS232 bisogna settare la velocità di trasmissione/ricezione dati. Questa velocità viene memorizzata qui, in forma

codificata. All'accensione il valore contenuto è 12, equivalente a un baud-rate di 19200. Si tratta del più alto valore standard supportato dallo Spectrum. Modificando questa variabile si può usare la RS-232 con velocità non-standard. Le applicazioni sono un po' limitate, ma se avete fantasia forse ne troverete qualcuna. Comunque eccovi la formula per inserire una baud rate non standard:

BAUD = (3500000 / (26 * baud-rate)) - 2

Ovviamente BAUD va inserito nelle locazioni relative con il solito metodo del byte più significativo per secondo.

23749. In questo byte è contenuto il numero di stazione nell'uso in rete locale. All'accensione vale 1.

23750. È il colore del BORDER durante le operazioni di ingresso-uscita. Inizialmente è settato a nero.

23751. Questi due bytes sono a disposizione della RS232 come area di lavoro. La locazione 23751 contiene i flag; la 23752 serve come buffer per un carattere in ingresso.

23753. Questa è invece l'area di lavoro dei microdrives. Contiene normalmente il numero dei settori. Ad esempio, quando si esegue un FORMAT, la variabile viene inizializzata a 255 e decrementata mano a mano fino a 0.

Quando invece si esegue un ERASE viene posta a 1275 e decrementata; in effetti l'ERASE procede in questo modo: vengono cercati all'interno del nastro tutti i settori con lo stesso nome e, in particolare, viene cercato un settore speciale, quello di fine file: questo contiene il numero di settori che compongono quel file (non importa se è basic, codice o un vero e proprio file di dati); una volta trovato il settore di fine file, lo Spectrum conta quanti sono i settori che ha trovato con il nome specificato, e subito dopo comincia a cancellare.

Nel caso non riuscisse a trovare il settore di fine file, lo Spectrum segna nella mappa tutti i settori con il nome interes-

sato, e li verifica 5 volte prima di cominciare a cancellare: ecco perché se dovete cancellare un file che non è stato chiuso occorre così tanto tempo.

23755. È usata dalla routine di controllo della sintassi, durante i controlli sulle linee che includono comandi esterni.

23757. Usata dalla rete locale, per memorizzare il numero della stazione ricevente e il segnale di handshaking, cioè l'OK alla trasmissione.

23758. È il numero della stazione a cui stiamo trasmettendo (sempre in rete locale).

23759. Questo invece è il numero della stazione che fa da sorgente (source).

23760. Qui viene memorizzato il numero progressivo del blocco dati inviato attraverso la rete locale.

23762. Quando è posto a 1 indica l'EOF della rete locale. È una cosa simile al blocco di fine file.

23763. Lunghezza del blocco dati trasmesso in rete locale.

23764. Qui è contenuto il checksum del blocco trasmesso dalla rete locale.

23765. Questo invece contiene il checksum dell'installazione del blocco dati, sempre della rete locale.

23766. Questa locazione e quelle che seguiranno fino alla 23772 hanno un uso un po' vario perché servono di volta in volta ai vari canali aperti. Se si sta lavorando con i microdrive, contiene il numero del drive chiamato in quel momento.

Se lavoriamo con la network, questa contiene il numero di stazione destinataria.

23768. Numero di canale logico che si sta utilizzando in operazioni di I/O.

23769. Identificatore del flusso al quale questo canale logico è assegnato (N, T, B, M, K, S).

23770. Lunghezza del nome del file.

23772. Qui c'è registrato l'inizio del nome del file, soltanto cioè due caratteri; il nome completo del file, del quale si stanno trattando le informazioni, è residente nell'apposito spazio riservato ai canali di comunicazione, dopo le variabili di sistema.

23774. Finito il blocco di variabili di sistema che servono canali di uso generale, da qui fino alla fine ci sono le variabili che regolano i microdrives.

Questa, in particolare, dovrebbe essere simile alla 23766, cioè una specificazione di file, però come funzionano i vari bytes (8) non è del tutto chiaro; orientativamente, sappiate che qui dentro ci deve essere il numero di microdrive con il quale lo Spectrum sta trattando.

23782. Questa è molto più chiara, e prevede molti utilizzi: contiene i codici dei vari tipi di file (ricordate come lo Spectrum gestisce l'header con il registratore a cassette...).

- 0 = Programma
- 1 = Array numerico
- 2 = Array stringa
- 3 = Bytes di codice macchina

23783. Memorizza quanti sono i dati da salvare (non importa se sono Array, Codice o Basic)

23785. Punto di inizio del programma o dei dati. Se state salvando è il punto dal quale lo Spectrum comincia a leggere i dati, se state caricando è il punto dal quale inizia a scrivere. Locazione di memoria da mandare a memoria.

23787. Lunghezza del programma o nome dell'array.

23789. Numero di linee usato per l'autoesecuzione.

23791. Numero di copie fatte da SAVE. Provate a indovinare cosa cambia questa variabile, se non ci riuscite sappiate che se POKeate 3 in questa locazione verranno scritte tre copie dello stesso programma nella cartuccia (potete fare da 1 a 255 copie!!!).

Questa possibilità è stata inserita per far sì che il tempo medio statistico di caricamento di un programma diminuisca: infatti se ci sono più copie di un programma su un nastro è più probabile che lo Spectrum lo carichi in minor tempo.

Da qui in poi comincia NORMALMENTE l'area di lavoro basic, ma quando utilizzate uno qualsiasi dei canali vengono memorizzate:

- 1) la mappa del microdrive
- 2) un'altra serie di variabili di sistema che sono intimamente legate al flusso aperto.
- 3) i dati relativi al canale aperto e i relativi checksum.

La parte più interessante è proprio quella relativa alle variabili che si creano quando si apre un file, in particolar modo verso il microdrive.

Orologio digitale

di Stefano Furlani

Sul numero di dicembre è stato presentato un programma in linguaggio macchina che visualizza un orologio digitale sullo schermo, lasciando il computer libero di lavorare ("Le interruzioni", SC n. 9 pag. 7). Nel testo è stato scritto che il timer sarebbe apparso premendo due tasti contemporaneamente: in realtà, chi ha caricato il programma si sarà accorto che l'ora restava permanentemente nell'angolo in alto a destra del monitor, il che in qualche caso può disturbare.

Ecco quindi una versione corretta del programma, che aggiorna l'orologio senza produrre messaggi al video, finché l'operatore non lo desidera (il listato si trova nella parte centrale della rivista).

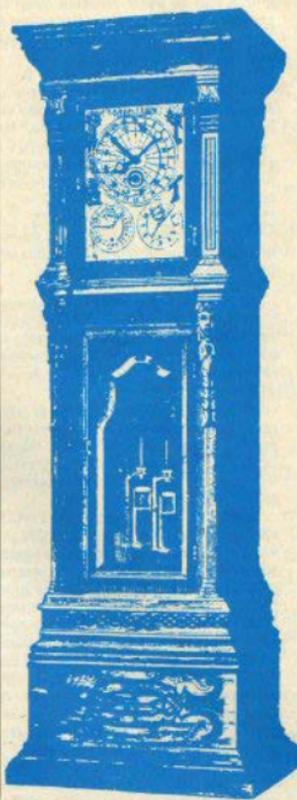
Per attivare l'orologio rimane sempre valida l'istruzione RANDOMIZE USR 65281 dopo la quale, per vedere che ora segna lo Spectrum, basta premere simultaneamente i tasti "1" e "2". Per impostare l'ora (ripetiamo per chi non avesse il n. 9):

POKE 65278, ore
POKE 65277, minuti
POKE 65276, secondi

mentre per disattivarlo (ripristina il modo interruzioni normale) date RANDOMIZE USR 65434.

Lo stesso programma può funzionare come il vecchio, tenendo cioè costante la presenza dell'orologio sullo schermo, dando semplicemente l'istruzione POKE 65331,0.

La stessa POKE con il valore 3, porterà alle condizioni originali.



Questa utility esegue una scansione completa del programma basic, linea per linea, calcolando alcuni parametri che possono risultare molto utili per il debugging, specialmente quando il programma è stato digitato da una copia su carta; bisogna ovviamente possedere un output di Checksum della versione originale, e la proposta dell'autore è appunto di standardizzarne l'uso.

Il programma, tutto in cod. macchina, risulterà all'occhio esperto più lungo dello stretto necessario; ciò è dovuto all'esigenza di renderlo autorilocabile. Infatti può essere caricato a qualsiasi indirizzo desiderato con comando diretto

LOAD "check" CODE ind
tenendo presente che occupa 152 bytes, e va eseguito con comando diretto RAND USR ind.

Ove necessario, RAMTOP va ovviamente impostato a ind-1.

L'output prodotto indica la lunghezza totale del programma in me-

Check-sum per il basic

di Giovanni Mellina

(list a pag. 40)

moria (occupazione in bytes) e, per ogni linea di programma:

— numero di linea
— lunghezza linea (*length text* + ENTER)

— *check-value*, ottenuto con SOMMA valori singoli bytes - INT (SOMMA/256)*256 + INT (SOMMA/256)

in certi casi ancora diminuito di 1.

L'output può essere ottenuto su stampante, sostituendo il valore 2 a ind+30 con il valore 3 (POKE ind+30,3). Il programma «CHECKLOAD» di cui pubblichiamo il listato esegue il caricamento ed il SAVE del cod. macchina all'indirizzo specificato in linea 20; dopo il basic trovato, come di consueto, il listato assembly.

Si tratta, come ammette l'autore, di una traduzione da un programma per il MICROPROFESSOR 2 (APPLE compatibile), per la divisione sillabica delle parole della lingua italiana.

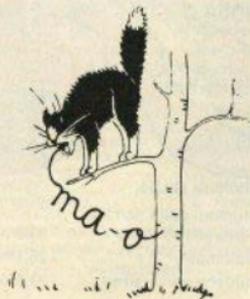
Il funzionamento è semplicissimo: il programma riceve in input una parola e, supponendo che quest'ultima appartenga all'insieme delle parole del lessico italiano, ne stampa la divisione in sillabe sullo schermo; è quindi ovvio che non viene fatto nessun controllo sulla correttezza della parola inserita. Controllo del resto fattibile, disponendo di qualche Megabyte di memoria di massa e di qualcuno che inserisca una per una tutte le parole di un qualsiasi dizionario... Per chi ne avesse la curiosità, ecco le principali regole di divisione sillabica della lingua italiana (Fonte: Corso di Documentazione Automatica, Facoltà di Scienze dell'Informazione di Milano):

1. *Consonante in mezzo a due vocali*: la consonante si unisce alla vocale seguente.

Sillabe

di Luca Ghisleni

(list a pag. 31)



2. *Vocale in mezzo a due consonanti*: la vocale si unisce alla consonante precedente (es.: *fel/li/ci/tà*).

3. *Consonante o gruppo di consonanti in testa alla parola*: si uniscono alla vocale seguente (es.: *prilmo, psi/che*).

4. *Gruppi di consonanti a due a due*: se le consonanti sono doppie, si divide a metà senz'altro. Altrimenti, si divide ancora a metà se le consonanti appartengono al gruppo B/D/N/S (Es.: *ab/norme, ab/dicare, ab/side*) o al gruppo C/M/N/S/T/Z (Es.: *tecnico, in/tenzione*), oppure sono le consonanti CQ (Es.: *ac/qua*).

Se le condizioni non si verificano si deve controllare la sillaba seguente, e formare due sillabe secondo la situazione (questo vale anche per i casi di tre consonanti vicine).

Notare che le parole più usate (si fa per dire...) che, secondo le regole, non potrebbero essere divise in sillabe sono: *tungsteno* e *feldmare-sciallo*.

Convenevoli

Innanzitutto, grazie a voi, che avete risposto numerosi. Nell'elaborazione dei dati abbiamo considerato le schede pervenute entro il 31 dicembre 1984, che sono circa 250.

Ci scusiamo se in alcuni punti il questionario non era chiarissimo; Comunque avete risposto con precisione e ne siamo soddisfatti.

Ringraziamo anche gli entusiasti, come quel Luca di Nonsisadove, che ci hanno snocciolato risposte sul tipo: "Per che cosa uso il computer?" "Per un casino di robe!" Oppure: "Quali riviste leggo?" "Quasi tutte!" Solo che questo tipo di risposte non ci tornava molto utile.

Come pure quelle puntigliose indicazioni di Nome, Cognome, Via e Numero Civico, cui non seguiva quella della città, del paese, della regione almeno. E non sono stati pochi.

Tuttavia, a conti fatti, l'immagine del Sinclairista che viene fuori da tutta la faccenda, ci sembra meglio definita e precisa. Ne terremo conto.

1. Diffusione per aree geografiche

La nascita di S.C. avviene di regola, non senza un considerevole travaglio, lungo l'asse ferroviario Milano-Brescia, interruzioni da neve permettendo. Da qui, la rivista si diffonde poi a tutto il resto d'Italia, gradualmente diradandosi con l'aumentare della distanza. Per cui i nostri lettori sono così distribuiti:

nord 54,9%
centro 28,6%
sud e isole 16,5%

2. L'età dei lettori.

meno di 15: 7,4%
15-18: 25,2%
19-24: 29,5%
25-35: 23,5%
più di 35: 14,4%

3. Il sesso.

Le lettrici sono l'1,27% del totale. Decisamente sembra che l'informatica non sia quella cosa che ti fa conoscere un sacco di ragazze!

I risultati del benchmark del lettore

a cura di Paolo Beneventi

Qui noi allora rivolgiamo un accorato appello alle Sinclairiste femminine: se ci siete, battete un colpo!

4. Occupazione.

Fra il 55,9% di studenti e lo 0,5% di pensionati, nella schiera dei nostri lettori ci sta di tutto. Manca solo la categoria dei religiosi, ma attiriamo la cosa alla loro naturale riservatezza, rifiutando di credere che così scarsa sia stata la nostra presa su conventi e parrocchie. Sappiamo per certo che i monaci benedettini di Cluny già da tempo lo minavano la parola "Sinclair" in 4 colori.

Per i giorni nostri, siamo fiduciosi.

studenti 55,9%
impiegati 1,5%
professionisti "informatici" 7,6%
disoccupati varie 4,9%
insegnanti artigiani 2,6%
ricercatori 2,6%
commercianti militari 1,3%
operai 0,5%
pensionati 0,5%

5. Titolo di studio.

Studenti delle scuole superiori 33,3%
diplomati 26,4%
studenti universitari 21,2%

laureati 12,1%
scuola dell'obbligo terminata o in corso 7%

6. Chi ha parenti che si occupano di informatica

Il 29% dei lettori. Fra questi 2/5 circa se ne occupano per lavoro.

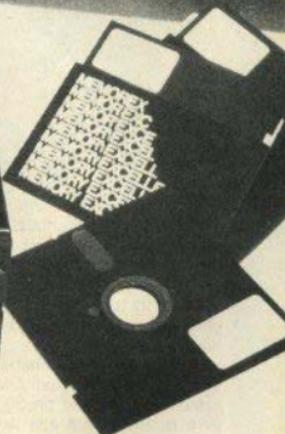
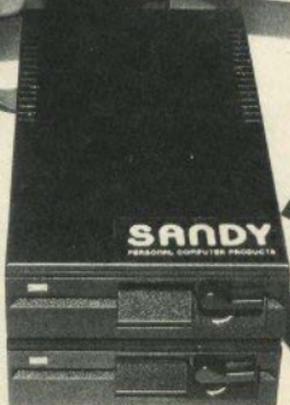
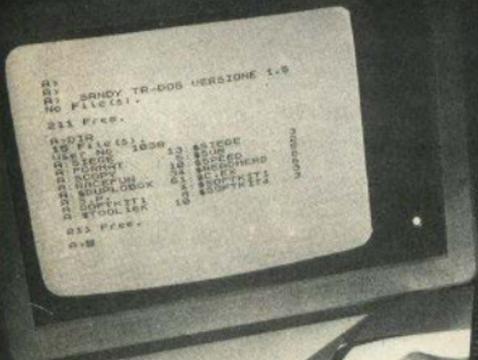
7. Mi occupo di informatica per:

hobby: 73,7%
lavoro e hobby: 24,1%



SANDY

PRODOTTI
PER HOME E
PERSONAL
COMPUTER



SINCLAIR ZX SPECTRUM & ACCESSORI

- QL L. 777
- SPECTRUM 48K: L. 395.000
- INTERFACE 1: inter RS232 indispensabile per il collegamento del microdrive.
- MICRODRIVE: drive per micro cartucce ottiche Sinclair.
- SUPERFACE: cart. vocale + gen. di suoni ampl. conno + interfaccia joystick e registratore.
- TAVOLETTA GRAFICA: consente di catturare immagini grafiche in alta risoluzione.
- TASTIERA: con pad. numerico più alloggiamento alim. ed eventuali interfaccia.
- MODEM: rivelazione strumento di comunicazione tramite linea telefonica.
- VENUTA PER CORRISPONDENZA PRESSO:

- EPROM PROGRAMMER: può programmare 2716/ 2732/ 2764/ 27128 completo di software. L. 270.000
- INTERF. RS232: adatto per collegare stampanti modem, plotter ect. L. 90.000
- INTERF. CENTRONICS: adatto per collegare qualsiasi stampante professionale. L. 120.000
- INTERF. JOYSTICK: programmi senza ausilio di software né hardware. L. 69.000
- JOYSTICK: L. 23.000
- ESPANSIONI 48K: L. 75.000

Per tutto il materiale non elencato (monitor, stampanti, software... ect) richiedere il catalogo.

IVA 18% ESCLUSA

NOVITÀ!!! FLOPPY DISK DRIVE PER SPECTRUM



CARATTERISTICHE PRINCIPALI

- Velocità da 3" e 5" da 100 a 900 bytes.
- Sistema operativo in non solo spazio in rom.
- Possibilità di collegare fino a quattro drive con una interfaccia (32 megabyte).
- Facile conversione di programmi. Modello da 100 bytes. L. 610.000

SANDY

PERSONAL COMPUTER PRODUCTS S.R.L.
Via Monterosa 22 Senago (MI) tel. 02-9869407

VENUTA DIRETTA PRESSO:
SANDY COMPUTER CENTER
VIA ORNATO 14 - TEL. 02-6473621
MILANO

BELLUNO - CIB. COMPUTERS P.zza S. Stefano, 1 tel. 0437-212204

NAPOLI - (LAPRETEL) Via Andrie, 71 tel. 081-657365

NOVARA - SYELDIS Via S.P. d'Assisi, 20 tel. 0321-27786

TRIESTE - C.I.S. GAGLIARDI Via Poate Res. 6 tel. 040-61692

SINCLAIR SONO MARCHI REGISTRATI
RESEARCH, L.T.O.



solo per lavoro: 2,2%

8. Lavoro/ho lavorato su

Che confusione quei termini: "mini", "micro", "personal"!

In realtà, definizioni così in bilico fra il tecnico e il merceologico sono una vera trappola, e male abbiamo fatto anche noi a non tenerne debitamente conto.

Abbiamo comunque appreso dell'esistenza di un consistente manipolo di lettori (dal 10 al 15%) che trovano evidentemente troppo brusco il passaggio dai grossi e seri computers che usano sul lavoro direttamente alle pantofole e allora, arrivati a casa, si rilassano alle tastiere "soft" degli Spectrum. Proprio quello che si chiama una *graduale disintossicazione*

9. Conosco bene i linguaggi:

BASIC	86,4%
FORTAN	11 %
COBOL	8 %
PASCAL	8 %
FORTH	4,7%
LOGO	2,1%
altri	13,6%
nessuno o non indicato	13,6%

10. Possiedo (o utilizzo)

Qui i dati parlano chiaro: il 94,9% degli utenti Sinclair possiede o utilizza uno Spectrum, solo l'11,5% lo ZX 81. Come è evidente, fra questi c'è chi li ha tutti e due. Poi, il 9,4% possiede anche computers di altre marche.

Pur con tutta la simpatia che riserviamo agli irriducibili ZXottantunisti, non possiamo perciò rispondere positivamente alle lamentele che ci sono giunte a proposito della graduale diminuzione dello spazio loro riservato sulla rivista: scontremmo quell'88,5% di lettori a cui — purtroppo per loro — i problemi del vecchio glorioso proprio non interessano. Anche perché ora dovremo fare spazio al QL.

11. Uso lo Spectrum soprattutto per:

giochi e svago	74 %
imparare a programmare	32 %

studio	27,4%
progr. di utilità e gestionali	26,9%
lavoro	12,8%
grafica	5,9%

12. È dotato dei seguenti accessori:

registratore	97 %
joystick	37,4%
monitor b/n	36,2%
stampante	29,4%
monitor colore	16,2%
microdrive	11,5%
penna ottica	5,9%
floppy disk-dr.	2,1%
tavoletta grafica	2,1%

13. Entro breve tempo prevedo di acquistare.

Qui scopriamo che una buona fetta di lettori, per il loro compleanno, hanno chiesto in regalo una stampante nuova. Anche noi!

stampante	32,8%
microdrive	20,4%
monitor colore	14,5%
joystick	8,5%
floppy disk drive	6,8%
penna ottica	5,1%
tavoletta grafica	3,8%
parlante Currah	1,3%
monitor b/n	0,4%
modem	0,4%

A proposito di modem, non mandate il prossimo numero: cercheremo di far salire questo misero 0,4%.

14. I programmi che utilizzo di più

Dall'esame dei programmi che vanno per la maggiore, se ne ricava che quasi tutti ci giocano, più della metà ci studia o ci lavora, tanti coi Sinclair si fanno gli affari loro (gestionali); e c'è anche un buon 30% che ci scrive (sarà la libidine dei tasti morbidi...).

giochi	82,1%
scientifici	57,9%
gestionali	45,1%
word-processing	29,8%

fra i giochi, le preferenze sono così distribuite:

arcades	27,6%
adventures	24,1%
strategia	17,8%
simulazione	6,9%
didattici	1,1%
tutti o vari	22,4%

15. Riviste di informatica che leggo.

Ma dove lo trovate il tempo di leggere tutta quella roba?

una o due:	17 %
da 3 a 5:	50,2%
da 6 a 9:	28 %
10 o più:	4,8%

16. Leggo libri di informatica:

regolarmente	38,5%
saltuariamente	54,3%
no o non rispondono	7,2%

17/18/19.

possiede tutti i numeri di S.C.	67,6%
sono abbonato è la prima volta che lo compro	14,5% 4,2%

20. Come ho trovato il n. 6 di S.C.

Dalle risposte arrivate, ne deduciamo che la nuova veste di S.C. fin dal suo debutto, è piaciuta al più. Tuttavia ci rendiamo conto che il questionario non era molto chiaro su questo punto e ha dato luogo a qualche equivoco.

21. La mia copia di S.C. viene normalmente letta da altri.

Più della metà dei Sinclairisti (il 54,5%) tiene la sua rivista tutta per sé. Tra gli altri:

da 1 persona	n. 31
da 2 persone	n. 31
da 3 persone	n. 16
da 4 persone	n. 6
da 5 persone	n. 3
da 6 o più	n. 4

22. Ho trovato più interessante...

Indice di gradimento:	
articoli	73,3%
software	72,8%
rubriche	60,8%

23. In particolare.

Program. assembly	16,6%
Totocalcio	13,6%
La posta	11 %
Viaggio nello Spectrum	10,6%
Prove su strada e novità	8,5%
Variabili di sistema	6,8%
Software matematico	6,4%



DIGITARE STANCA



DIGITARE STANCA!

I programmi più interessanti spesso sono molto lunghi, un listato pubblicato è faticoso da leggere...

Sinclair Computer vi offre un'alternativa: le cassette con tutti i programmi pubblicati sulla rivista.

Ogni nastro contiene il software di un numero di Sinclair Computer, a un prezzo incredibilmente basso: solo 5.800 lire (+ 1.000 lire per spese di spedizione).

Riceverete le cassette direttamente a casa vostra, utilizzando il coupon qui a fianco.

DIGITARE STANCA è un'iniziativa



systems

Desidero ricevere le cassette con il software pubblicato sui seguenti numeri di Sinclair Computer:

.....
importo L.
spese di spedizione L. 1.000
Totale L.

ho versato l'importo sul c/c postale n. 37952207 (allego fotocopia della ricevuta di versamento)
 accludo assegno non trasf.

n. (banca)
intestato a SYSTEMS Editoriale
V.le Famagosta 75, 20142 MILANO

nome
cognome
via
CAP/città

Ritagliare e spedire in busta: a Systems Editoriale v.le Famagosta 75, 20142 Milano.

ABBONATEVI A SINCLAIR COMPUTER



Un'iniziativa condotta con la nota rivista Computer



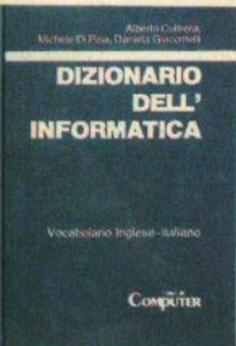
PROGRAMMO IN BASIC

Il linguaggio del futuro in un manuale rapido e completo di Cizio Merli
pagg. 224 (L. 9.000)

Il Basic, attualmente il linguaggio più conosciuto - adatto all'utilizzo su qualunque tipo di macchina e in particolare sui personal e gli home-computer - può essere appreso in poche ore con l'aiuto di questo agile manuale.

I volumi, che sono comunque in vendita nelle migliori librerie di tutta Italia, possono anche essere richiesti direttamente all'Editore.

Importante: l'ordine minimo dovrà essere di L. 15.000.



COME SCEGLIERE UN COMPUTER

Guida pratica per l'acquisto di un mini o di un micro computer professionale di Michele Di Pisa
pagg. 160 (L. 6.000)

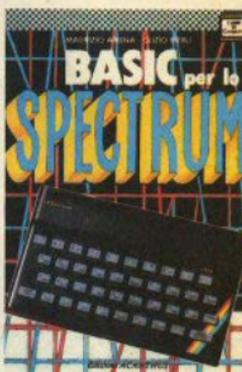
Quale modello scegliere tra gli oltre 600 computer commercializzati in Italia? La conoscenza delle caratteristiche delle varie macchine è indispensabile. Con un approccio a "menu" l'Autore vuol essere guida proprio in questa fase.



UTILITY E ROUTINE PER IL COMMODORE 64

di Girolano Rossi
pagg. 192 (L. 9.000)

L'esecuzione di una istruzione BASIC può richiedere diverse centinaia di passi di programmi in linguaggio macchina. La dimensione dei programmi è ciò che intimidisce maggiormente l'utilizzatore medio di Commodore: aiutato da questo testo chiunque potrà affrontare senza problemi il processo di scrittura di un programma.



BASIC PER LO SPECTRUM

di Maurizio Ariani e Cizio Merli
pagg. 192 (L. 9.000)

Un libro per quanti hanno acquistato il computer ZX Spectrum della Sinclair e intendono sfruttarne appieno tutte le capacità, dall'hardware alla programmazione in assembly (linguaggio macchina).



Edizioni ACANTHUS

VIALE GRAN SASSO, 23 - 20131 MILANO

Inviatemi i seguenti volumi:

Titolo	quantità	prezzo unitario
_____	_____	_____
_____	_____	_____
_____	_____	_____
spese postali		L. 2.000
	totale	L.

Pagherò contrassegno (il dovuto più L. 2.000 per contributo spese postali) al ricevimento. Potrà restituire i libri entro 8 giorni se non saranno di mio gradimento e avere il rimborso immediato.

COGNOME _____

NOME _____

VIA _____ N. _____

C.A.P. _____ CITTÀ _____

FIRMA _____

DATA _____





Un tasto per volta

Tutta la tastiera dello Spectrum

Y

NORMALE

Istruzione RETURN. Per avere un significato, deve trovarsi al termine di una subroutine, che viene chiamata da GOSUB, di cui RETURN è il naturale complemento. Come vedremo, GOSUB xxx provoca un salto all'istruzione xxx, memorizzando automaticamente il numero della linea da cui è partito.

Il programma quindi prosegue l'esecuzione dalla linea xxx, fino a che non incontra un RETURN, che provoca un salto indietro, all'istruzione immediata successiva al GOSUB.

Potete utilizzare più statements di GOSUB in successione: lo Spectrum è in grado di ricordare i punti di ritorno di ben 255 subroutines contemporaneamente. L'area in cui le informazioni per i RETURN vengono memorizzate si trova nella parte più alta della RAM, appena prima di RAMTOP (quindi subito "sotto" gli UDG, a meno di modifiche alla RAMTOP stessa). Trattandosi di un'area dinamica, la forzatura di un return risulta un'operazione piuttosto difficile da basic.

Nell'improbabile ipotesi che riusciate a saturare lo spazio disponibile per la GOSUB stack, ottenete un messaggio 4 Out of memory

SYMBOL SHIFT

Operatore logico AND. Nel programma dei principianti gli operatori logici compaiono raramente, forse per una

sottovalutazione delle possibilità che offrono; per questo motivo dedichiamo un ampio spazio a AND e OR.

Il significato più semplice di questa istruzione è quello letterale, cioè "e" (congiunzione).

Vediamo subito qualche esempio:

```
100)IF a<10AND b<10THEN GO TO i
istruzione
```

Nel caso che sia *a* che *b* siano minori di 10, verrà eseguita l'istruzione che segue il THEN. Questo uso, di immediata comprensione, non è però il più interessante; ponete di dover affrontare le seguenti condizioni:

```
50 IF a<10 THEN GO TO 100
60 IF a=10 THEN GO TO 200
70 IF a>10 THEN GO TO 300
```

Non disponendo, sullo Spectrum, dell'istruzione ON... GOTO, si può risolvere il problema molto elegantemente nel modo che segue:

```
50 GO TO (100 AND a<10)+(200 AND
a=10)+(300 AND a>10)
```

Ciascuna espressione tra parentesi viene esaminata dal computer come test *vero/falso*; i termini di sinistra, numeri, sono necessariamente *veri*, tra i termini di destra ve ne sarà di volta in volta solo uno *vero*.

Alle espressioni contenenti i termini *falsi* viene assegnato valore logico zero, e ciò permette di risolvere univocamente l'istruzione di salto; per esempio, se *a* = 20 (quindi maggiore di 10), abbiamo:

```
50 GO TO (0)+(0)+(300)
```

Vediamo un altro caso, un po' più complicato:

```
50 IF b<10 THEN GO TO 100
60 IF b=10 THEN GO TO 200
70 IF b<20 THEN GO TO 300
```

Com'è evidente, *b* può essere contemporaneamente minore di 10 e di 20, o può essere uguale a 10 — e quindi ancora minore di 20. Nella forma scritta, le istruzioni sono disposte nel giusto ordine: la terza condizione verrà raggiunta se *b* è compreso tra 11 e 20.

Tuttavia, se provassimo a riscrivere quanto sopra usando gli operatori logici, ci troveremo in difficoltà:

```
50 GO TO (100 AND b<10)+(200 AND
b=10)+(300 AND b<20)
```

può sembrare corretta, ma se assegnate per es. *b* = 10 vedete subito che si realizza la verità di due condizioni

```
50 GO TO (0)+(200)+(300)
```

e il programma cercherà di eseguire GOTO 500. La forma corretta è la seguente:

```
50 GO TO (100 AND b<10)+(200 AND
b=10)+(300 AND (b<20 AND b>10))
```

Le parentesi interne nell'ultima condizione non sono strettamente indispensabili: le abbiamo aggiunte per chiarezza espositiva; se avete dei dubbi sulla priorità di esecuzione delle varie scelte, vi consigliamo di scrivere tutte le coppie di parentesi opportune: come al solito, è meglio abbondare.

Ecco qualche altra situazione d'uso

di AND:

```
50 PRINT AT (n AND x>15), (n AN
D x<10);a#
60 PLOT k,(7 AND x=13)+(37 AND
x<13)
70 PAUSE (200 AND t=1)+(400 A
ND t=13)
80 LET x=x+(10 AND h<=k)-(7 AN
D h>k)
```

(verificate gli effetti). Oltre ai valori numerici, si possono trattare stringhe: quando il risultato dell'operazione logica è vero, il contenuto delle parentesi diventa la stringa che precede l'AND.

```
50 PRINT ("Troppo grande" AND
x>y)+("Troppo piccolo" AND x<y)+
("Giusto" AND x=y)
```

risolve in una sola linea di programma il classico giochetto "indovina un numero". Altro esempio:

```
50 PRINT a#(n);TAB 12;("** AND
x(n)<=15);TAB 18;x(n)
```

che tabula stringa e numeri associati, evidenziando con un asterisco tutti quelli che superano un valore prefissato.

AND ha priorità più alta di OR, come vedremo meglio tra poco, nel paragrafo dedicato a quest'ultimo: cioè viene affrontata per prima, indipendentemente della sua posizione (v. OR).

CAPS SHIFT

«Y» maiuscola.

MODO «E»

Istruzione STRS. Richiede per argomento un numero o una variabile numerica, che trasforma in una variabile stringa.

L'utilità di trattare dati numerici come se fossero parole si rivela soprattutto in fase di stampa: grazie a STRS e a qualche artificio è infatti possibile simulare le PRINT USING del basic Microsoft, cioè le maschere di stampa che consentono di tabulare in formati regolari i risultati di un'elaborazione.

Le istruzioni che seguono sono un piccolo assaggio di quello che è possibile fare: il formato dell'output andrà studiato caso per caso; qui si normalizzano a tre le cifre decimali.

```
50 LET q#="STR$(1+(q-INT q))!"+
000"
60 LET q#="STR$ INT q*,"+q#(3
TO 5)
```

SYMBOL S. in «E»

Parentesi quadra aperta «[». Ha un uso solo grafico, e non può essere usata nelle espressioni in luogo della parentesi tonda.

U

NORMALE

Statement IF. Cioè «se»: «IF (condizione) THEN (scelta)».

Inscindibilmente legato a THEN, senza cui non può essere utilizzato, IF è una delle istruzioni più importanti del basic, poiché ci consente di prendere decisioni nello svolgimento del programma, condizionate al verificarsi di determinate istruzioni; ne abbiamo visto l'uso più sopra, in alternativa a AND.

Un errore frequente del neofita è l'aggiunta dopo il THEN di altre situazioni, oltre alla prima, anche se queste non devono dipendere dal realizzarsi della condizione:

```
IF (condizione) THEN (istruz.a):
(istruz.b): (istruz. c)
```

In questo modo tutte le operazioni (a,b,c) vengono eseguite se (e solo se) la condizione è vera; se (b) e (c) devono essere eseguite sempre, occorre metterle in una linea di programma successiva. Sarebbe possibile mantenerli nella linea dell'IF se si disponesse dell'opzione ELSE («altrimenti»), che fa parte, come al solito, del set di istruzioni Microsoft e simili.

Non è comunque il caso di sentirsi troppo frustrati: ELSE è certamente utile, ma è molto raro imbattersi in problemi di programmazione non risolvibili senza di essa.

SYMBOL SHIFT

Operatore logico OR. Fa il paio con l'AND di cui abbiamo ampiamente discusso prima. Segue la stessa sintassi, con la differenza che mentre AND viene usato per addizioni/sottrazioni, OR si utilizza per moltiplicazioni e divisioni. Il suo significato è «oppure», o anche «a meno che...». Nell'esempio

```
50 IF a<10 OR b<10 THEN GO TO
istruzione
```

quanto segue il THEN viene elaborato se almeno una delle due condizioni è vera. Fate attenzione, nel programmare, a non confondere l'OR normale o inclusivo dall'OR esclusivo, solitamente sigla-

to XOR, presente nello Spectrum solo come istruzione assembly (c'è, invece, nel SuperBasic QL): XOR consente l'esecuzione solo se una o l'altra delle due condizioni è vera, e salta se sono entrambe vere o entrambe false.

Se avete bisogno di un OR esclusivo, potete combinare AND e OR; ecco l'esempio di prima con una simulazione di XOR:

```
50 IF (a<10 OR b=10) OR (a=1
0 AND b<10) THEN GO TO istruzio
ne
```

Per comprendere meglio il funzionamento di OR come operatore logico, confrontate nella tabella la differenza da and:

AND (numero) vero	=	(numero)
AND (numero) falso	=	0
OR (numero) vero	=	1
OR (numero) falso	=	(numero)

Vediamo un esempio:

```
50 LET resto=resto*(10 OR a<b)
```

Se (a < b) è vero, all'espressione tra parentesi viene assegnato valore 1, e quindi il valore della variabile in assegnazione non cambia:

```
LET resto = resto * (1)
```

Se (a < b) è falso, l'espressione tra parentesi assume il valore che precede l'OR:

```
LET resto = resto * (10)
```

La variabile cioè viene moltiplicata per 10, «a meno che» a sia maggiore b, caso in cui resta invariata. Usando AND in questa situazione, nel caso di condizione falsa avremmo ottenuto l'azzeramento della variabile: una necessità anche questa che si presenta spesso. Un altro esempio:

```
50 LET score=score+(10 OR h<5)
*(10 OR h<10)*(10 OR h<15)
```

questo, che potrebbe rappresentare l'aggiornamento del punteggio di un gioco, funziona così: se h è minore di 5, si ha il vero su tutta la linea, e lo score non cambia:

```
LET score = score * (1)*(1)*(1)
```

Con h compreso, per esempio, tra 10 e 15, è vero solo la terza condizione, e il punteggio viene moltiplicato per 100:

```
LET score = score * (10)*(10)*(1).
```

Associando AND e OR bisogna prestare attenzione alle priorità: nel caso di

```
50 IF x<5 OR y<5 AND z<5 THEN
GO TO istruzione
```

lo Spectrum risolve prima l'AND, come se ci fossero le parentesi così:



```
50 IF x<5 OR (y<5 AND z<5) THE
N GO TO istruzione
```

Per alterare l'ordine di esecuzione, basterà ovviamente inserire altre opportune parentesi:

```
50 IF (x<5 OR y<5) AND z<5 THE
N GO TO istruzione
```

CAPS SHIFT

«U» maiuscola.

MODO «E»

Istruzione CHR\$. Richiede come argomento un numero compreso tra 0 e 255, e restituisce l'istruzione corrispondente al quel codice; "istruzione" dovete qui intenderla nel senso più ampio:

da 0 a 31 caratteri di controllo non stampabili

da 32 a 127 set di carattere secondo lo standard ASCII, con l'eccezione dei cod. 96 e 127; il set comprende l'alfabeto maiuscolo e minuscolo, i numeri, punteggiatura, segni algebrici principali e alcuni segni grafici

da 128 a 164 caratteri grafici speciali; i primi 16 già definiti, gli altri (i ben noti UDG) definibili dall'utente

da 165 a 255 a ciascun codice corrisponde un'istruzione basic.

Gli usi di CHR\$ sono svariati, e molto diversi tra loro in particolare, le possibilità di utilizzo con il primo gruppo di codici andrebbero esaminate singolarmente (lo faremo presto).

CHR\$ è indispensabile con le stampanti, per inviare i caratteri di controllo che definiscono il formato di stampa; per esempio, con le istruzioni che seguono vengono eseguiti i listati per Sinclair Computer (su Epson FX-80, via RS232 dell'interface 1):

```
999B CLEAR #: FORMAT "b",9600: F
ORMAT "t",9600: OPEN #4,"b": OPE
N #5,"t"
```

```
9999 PRINT #4;CHR$ 27;"A";CHR$ 1
1;CHR$ 27;"E";CHR$ 27;"B";CHR$ 2
7;"1";CHR$ 18;CHR$ 27;"Q";CHR$ 4
2: LIST #5
```

La serie di CHR\$ (che va definita consultando il manuale della stampante) riguarda: l'interlinea, il modo di stampa evidenziato, la doppia battuta di stampa, il margine sinistro e il margine destro.

SYMBOL S. IN "E"

Parentesi quadra chiusa "I". Ha un uso solo grafico, e non può essere usa-

ta nelle espressioni matematiche in luogo della parentesi tonda.

1

NORMALE

Statement INPUT. Serve per comunicare dati al computer, solitamente attraverso la tastiera. Nella sua forma più semplice è seguito soltanto dal nome della variabile (numerica o stringa) che deve essere assegnata con il dato in ingresso:

```
INPUT x
INPUT x$
```

Nel primo caso potete rispondere con un numero qualsiasi o con il nome di un'altra variabile numerica già definita in precedenza; per fermare il programma, digitare STOP (S.SH + "a").

Nel secondo caso, il cursore in attesa del dato appare tra le virgolette; si può rispondere con qualsiasi cosa, dal semplice ENTER, che assegna una stringa nulla, a una sequenza lunga quanto volete (in teoria, in cui sono ammessi anche numeri e nomi di comandi basic; spetterà poi alle istruzioni successive verificare l'ammissibilità del dato. Per fermare il programma, si usa ancora STOP, ma bisogna portarlo a sinistra degli apici con il cursore (C.SH. + "5"), o cancellare gli apici stessi.

```
Un terzo formato di INPUT è
INPUT LINE a$
```

che serve ancora per introdurre stringhe alfanumeriche, ma non fa comparire gli apici e non può essere fermato nel modo detto.

Per avere uno "STOP in INPUT" (messaggio HV), si deve dare C.SH. + "6".

INPUT può essere correata da un messaggio che specifica il tipo di dato da introdurre:

```
INPUT "Importo?"; lire
e si possono richiedere più dati con una sola istruzione:
```

```
INPUT "a'b'c'd'e".... (etc.)
INPUT "Data di nascita: " " " giorno ";
```

```
gg "mese"; mm "anno"; aa
```

L'introduzione dei dati può però avvenire anche tramite periferiche dalla tastiera, quali i microdrives o la porta RS232 (alla quale sarà collegato, per es., un modem). Il funzionamento di INPUT è lo stesso: a ciascun dato acquisito deve venire associato il nome di una variabile. Si modifica la sintassi, poiché dovremo specificare, di volta in volta, qual'è il canale di comunicazione utilizzato e a quale dispositivo è associato

(cfr. "Accesso random ai mdrives"; in questo stesso numero). Es.

```
50 OPEN #7;"a";!;"database"
60 INPUT #7;d1;d2;d3
```

SYMBOL SHIFT

Statement AT. È un'istruzione complementare di PRINT (non può essere usata da sola), e serve per stampare in posizioni prefissate dello schermo.

I due argomenti che richiede, separati da una virgola e seguiti da un punto e virgola, sono, nell'ordine, la linea e la colonna di stampa, e avremo perciò un range utile ben definito: da 0 a 21 e da 0 a 31. Perciò.

```
PRINT AT 7,9; "0"
```

stampa uno "0" sull'ottava riga e sulla decima colonna (la numerazione di righe e colonne inizia da 0). AT può servire anche con INPUT, con la stessa sintassi; cambia in questo caso il riferimento delle righe: INPUT AT 0,0 stampa all'inizio della 23esima riga, cioè la prima delle due righe di editing.

CAPS SHIFT

"I" maiuscola.

MODO "E"

Istruzione CODE. È l'inverso di CHR\$: richiede come argomento una stringa (racchiusa tra apici), e restituisce il codice corrispondente:

```
PRINT CODE "a"; CODE "d"; CODE
"E"; CODE "GO TO"
```

darà, nell'ordine: 67, 100, 69, 236; il "GO TO" è, naturalmente, l'istruzione reale, e non una trascrizione. Per ottenere un comando basic, che richiede il cursore in "K", all'interno di una stringa, dove il cursore è in "L" o "C", basta digitare prima un THEN (S.SH. + "G"), che poi si cancellerà, arretrando con C.SH. + "5".

Se la stringa fornita è più lunga di un byte, il risultato è il codice del primo carattere; non dimenticate che il "GO TO" visto prima, o gli altri comandi basic, anche se esplicitati in una parola, sono tutti lunghi un byte. Se la stringa è nulla ritorna 0.

L'uso di CODE in un programma dipende a volte dalle preferenze dell'autore: scrivere

```
IF INKEY$ = "7" THEN GO TO 1000
oppure
```

```
IF CODE INKEY$=55 THEN GO TO
1000
```

è la stessa cosa; peraltro, se le condizioni da esaminare sono molte e complesse, può essere più agevole trattare numeri.

CODE ha un altro uso, completamen-

te diverso, associato a SAVE, LOAD, VERIFY (sia con il registratore che con l'interfaccia 1 e i microdrives), definendo il salvataggio di blocchi di bytes in forma sequenziale (cioè semplicemente uno dopo l'altro, qualunque cosa contengano). Es.

SAVE "codici" CODE 57800, 3250

Con SAVE sono necessari due argomenti, il primo che specifica la locazione di inizio dell'area di memoria da salvare, il secondo che indica la lunghezza dell'area stessa. Con LOAD questi parametri non sono indispensabili, a meno che non si vogliano caricare i dati in una zona della RAM diversa da quella originaria.

SYMBOL S. IN "E"

Istruzione IN. Misteriosa per la maggior parte degli utenti, a causa delle spiegazioni piuttosto sibilline fornite dal manuale, è destinata (insieme alla omologa OUT) principalmente alla gestione di hardware esterno; sul connettore per le espansioni sono presenti tutte le linee del bus indirizzi del microprocessore, da A0 ad A15. Un dispositivo esterno invia dei valori logici, che il computer può rilevare e interpretare.

Un utilizzo tipico è quello delle interfacce per joystick: l'interfaccia 2 della Sinclair utilizza per esempio i primi 5 bit degli indirizzi 61438 e 63486, rispettivamente per il primo e per il secondo joystick.

Il programmino che segue, ripreso dal foglietto di Istruzioni dell'Interf. 2, spiega come possono venire usati i valori letti da IN.

```

50 LET x=0: LET y=80
60 GO SUB 1000
70 IF fire=1 THEN PLOT x,y
80 GO TO 60
1000 LET a=IN 61438
1010 IF a>127 THEN LET a=a-128
1020 IF a>63 THEN LET a=a-64
1030 IF a>31 THEN LET a=a-32
1040 IF a>15 THEN LET a=a-16: I
F x<255 THEN LET x=x+1
1050 IF a>7 THEN LET a=a-8: IF
x>8 THEN LET x=x-1
1060 IF a>3 THEN LET a=a-4: IF
y<175 THEN LET y=y+1
1070 IF a>1 THEN LET a=a-2: IF
y>8 THEN LET y=y-1
1080 LET fire=a
1090 RETURN
  
```



NORMALE

Istruzione POKE. Famigerato comando, in viso spesso sia ai neofiti che agli esperti, anche se per motivi diversi: ai primi, per la difficoltà di comprenderne il significato, ai secondi per la noia insita nell'uso prolungato.

POKE serve per scrivere in una locazione di memoria. Le locazioni possono venire immaginate come una fila di caselle, numerate progressivamente: lo Z80, il microprocessore a 8 bit dello Spectrum, può indirizzare 65536 caselle, da 0 a 65535.

L'operazione

POKE n,x

in cui il range utile per i due argomenti è da 0 a 65535 per n e da 0 a 255 per x, può essere eseguita su tutte le locazioni, senza che il computer dia messaggi di errore, ma per le prime 16384 non ha effetto, poiché questa è l'area del sistema operativo e del basic che, come sappiamo, sono residenti su memoria a sola lettura, non modificabile.

Gli usi possibili di POKE sono moltissimi, e possiamo fare solo qualche esempio:

a) scrivere nella memoria di schermo (da 16384 in poi, per 6912 bytes): di scarsa utilità, a causa del modo in cui è gestito il display-file; ci sono sistemi più pratici (PLOT, etc.).

b) nel buffer stampante (23296-23551) si possono caricare brevi routines in c.m., a patto ovviamente che la stampante non debba venire usata; spesso si scrive in qualche locazione di quest'area quando occorre passare un dato dal basic al c.m. o viceversa.

c) nelle variabili di sistema (23552-23734): una parte di esse, segnata con una X sul manuale, non può essere modificata senza rischiare il «crash» o l'autoretro; nelle altre si possono eseguire POKE a piacere, come è stato dettagliatamente spiegato sul n. 06/07, per ottenere una quantità di «effetti speciali»: forzare un GO TO, cambiare il clic dei tasti, modificare gli attributi di stampa, etc.

d) nell'area basic, si eseguono POKE solitamente solo nella prima linea di programma, l'unica le cui locazioni siano facilmente identificabili; altrove, occorre un programma monitor e una buona pratica

e) decisamente sconsigliabile è la manipolazione dell'area di lavoro del

calcolatore, da VARS (inizio area variabili) a RAMTOP, a meno che non siate veramente esperti.

f) l'ultima parte della memoria, da UDG a P-RAMT, è invece normalmente gestita con POKE: cfr. sul n. 10 quanto scritto per il tasto «9». Cfr. anche S.C. n. 05, pag. 06.

SYMBOL SHIFT

Punto e virgola «;». Oltre ad essere un normale segno di punteggiatura, il punto e virgola è fondamentale nella sintassi di molte istruzioni: viene usato con OPEN, MOVE, ERASE, FORMAT, SAVE/LOAD/MERGE/VERIFY (con i microdrives), AT, TAB, PRINT, LIST.

CAPS SHIFT

«O» maiuscola.

MODO «E»

Istruzione PEEK. È l'operazione inversa di POKE: deve essere seguita da un solo argomento, compreso tra 0 e 65535, e restituisce il valore decimale contenuto nella locazione indirizzata da quel numero. Come per POKE, gli usi sono molteplici.

La lettura della ROM è difficilmente utile (almeno in questo modo, a singole locazioni). In qualche caso, può essere usata per leggere un pixel del video (ma è più pratica POINT - cfr. sul n. 10). Importante e molto frequente la lettura di variabili di sistema; ricordate che in quelle di 2 bytes il più significativo è il secondo, per cui il valore che contengono si ottiene con

PEEK 23xxx + 256 * PEEK (23xxx + 1)

Cfr. anche S.C. n. 05, pag. 6.

SYMBOL S. in «E»

Istruzione OUT. Operazione inversa di IN: necessita di due argomenti, il primo indicante l'indirizzo al quale inviare un determinato valore, il secondo con il valore da spedire.

IN e OUT si possono paragonare, rispettivamente, a PEEK e POKE; in che modo un indirizzo dato a OUT venga interpretato dallo Spectrum «dipende dalla configurazione» (dice il manuale).

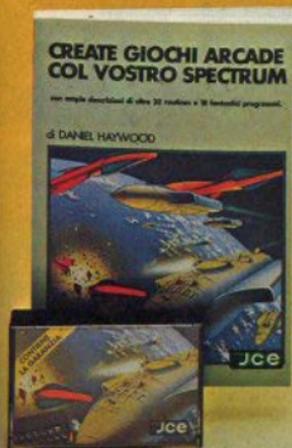
Spesso indirizzi diversi indicano la stessa configurazione; abbiamo consultato una ventina di libri sullo Spectrum, senza trovare soddisfacenti informazioni sulla mappatura di questi indirizzi (gli autori danno normalmente per scontato che chi legge sia esperto di hardware). Documentazioni migliori dai lettori saranno ben accette.



LIBRI

edizioni

Jce



CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM

di DANIEL HAYWOOD

Il volume descrive dettagliatamente tutte le tecniche di stesura di giochi ARCADE, partendo dalla lettura della tastiera e toccando la definizione grafica, l'impiego del suono e l'uso degli operatori logici, per migliorare la qualità dei programmi. Altri argomenti esaminati in dettaglio sono l'animazione degli oggetti, lo scrolling dello schermo e l'impiego dei comandi PEEK e POKE per il loro uso più corretto. Il tutto accompagnato da 18 programmi la maggior parte dei quali o sono inediti, oppure riguardano versioni migliorate di games di grande successo come "INVADERS" e "BOMBER". I più interessanti sono stati registrati sulla cassetta allegata al volume al fine di farvi risparmiare ore di digitazione.

 Pag. 118
 Libro più cassetta

Cod. 9003

L. 25.000

BASIC & FORTRAN PER SPECTRUM

di S. J. WAINWRIGHT e A. GRANT

Questo libro può essere utilizzato per imparare sia il FORTRAN che il BASIC, od anche per apprendere entrambi i linguaggi contemporaneamente sul vostro SPECTRUM. La cronologia dei linguaggi dimostra quindi che il FORTRAN occupa una posizione di rilievo, soprattutto per ciò che concerne la programmazione di problemi scientifici e matematici. Mentre il BASIC è il più diffuso dei linguaggi di programmazione per microcomputer.

In questo libro e nella cassetta allegata è stato inserito un interprete FORTRAN per lo SPECTRUM nella massima configurazione (48k) che però con alcune piccole modifiche chiaramente indicate nel testo, potrà essere utilizzato con la versione inespansa ed anche sul SINCLAIR ZX 81.

Questo interprete vi aiuterà subito a comprendere i fondamenti della programmazione in FORTRAN, rendendo possibile la creazione di programmi che vi daranno una visione ben precisa delle potenzialità caratteristiche di questo linguaggio.

 Pag. 88
 Libro più cassetta

Cod. 9007

L. 25.000

IMPARIAMO IL PASCAL SUL NOSTRO COMPUTER

di JEREMY RUSTON

Questo libro si rivolge a chi desidera conoscere il PASCAL ed apprendere l'uso in modo semplice e lineare, è quindi adatto anche a chi è alle prime armi nel campo dell'informatica.

Nel libro sono riportati i listati di due programmi compilati per tradurre le istruzioni PASCAL in BASIC, questo consente al lettore di provare direttamente programmi in PASCAL, sul suo personal computer senza dover affrontare la spesa di un vero compilatore PASCAL.

Il primo compilatore è scritto in Basic MICROSOFT, quindi è adatto ai personal computer IBM PC; IBM compatibili, OLIVETTI M. 10 - M. 20 - M. 21 - M. 24 - HP 150. Il secondo è scritto in Basic SINCLAIR per lo ZX SPECTRUM ed è fornito su cassetta software allegata al libro.

 Pag. 112
 Libro più cassetta

Descrizione	Cod.	Q.tà	Prezzo unitario	Prezzo Totale
CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM Libro più cassetta	9003		L. 25.000	
BASIC & FORTRAN PER SPECTRUM Libro più cassetta	9007		L. 25.000	
IMPARIAMO IL PASCAL SUL NOSTRO COMPUTER Libro più cassetta	9800		L. 25.000	

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

Nome

Cognome

Via

Città

Data C.A.P.

SPAZIO RISERVATO ALLE AZIENDE - SI RICHIEDE L'EMISSIONE DI FATTURA

 Partita I.V.A.

PAGAMENTO:

- Anticipato, mediante assegno bancario o vaglia postale per l'importo totale dell'ordinazione.
- Contro assegno, al postino l'importo totale
- AGGIUNGERE: L. 3.000 per contributo fisso spedizione. I prezzi sono comprensivi di I.V.A.

Via G. Verdi, 23/25



SI ACCETTANO FOTOCOPIE DI QUESTO MODULO D'ORDINE

dall'alto:
HERO,
Finders Keepers,
Chiller,
Beach-Head.

CHILLER
Mastertronic
Spectrum 48K.

Chiller è un gioco multischermo del tipo "Mantic Miner". Vi sono infatti 6 schermi differenti in ognuno dei quali vi è una differente scenografia, differenti oggetti da prendere, differenti nemici.

Scopo dichiarato è raggiungere la propria amata, rapita e custodita in una casa stregata (l'ultimo schermo), per liberarla dagli zombie, cioè dai "merli che camminano". Per accedere allo schermo successivo bisogna recuperare tutti gli oggetti luccicanti presenti, evitando il contatto con gli zombie o con funghi velenosi e pipistrelli.

Da notare che, a differenza di molti altri giochi, il contatto con un nemico non uccide istantaneamente, ma provoca una diminuzione della nostra energia (mostrata da un apposito indicatore in alto sul video), che può però essere ripristinata raccogliendo, ovvero mangiando, del cibo presente sullo schermo.

Un'altra differenza di Chiller rispetto a tanti altri programmi di questo genere è il basso numero di schermi, che però sono molto dettagliati graficamente; evidentemente il programmatore non ha voluto "pigriare" nella memoria del computer troppi schermi stilizzati ed ha preferito poche buone scenografie.

L'animazione non è velocissima (per questo non stanca troppo la vista del giocatore) e risulta relativamente fluida, senza perdite di colori dietro ai caratteri, come invece avviene, in altri giochi di recente produzione.

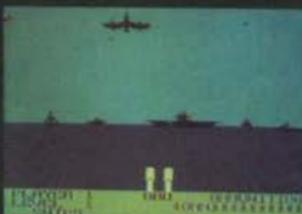
Inizialmente ci viene presentato un menù per selezionare il livello di gioco, i controlli (tastiera o joystick) e il modo dimostrativo.

I comandi sono elementari: sinistra, destra e salto.

Grafica: buona la scenografia, stilizzati i caratteri animali.

Comandi: molto semplici e pronti nella risposta.

Giocabilità: inizio facile con difficoltà crescente. Buona.



a cura
Luigi

ENSIO

H. E. R. O.
Activision
Spectrum 48K.

Durante la cerimonia di inaugurazione delle Olimpiadi di Los Angeles abbiamo assistito al prodigioso volo di un «parà» munito di «zaino a razzo». In questo gioco il protagonista è un milite che deve sfruttare questo futuristico mezzo di locomozione per salvare vite umane.

Il gioco si articola in più fasi, precisamente sono 20, di progressiva difficoltà. Sono disponibili 7 livelli di gioco, corrispondenti a 7 diverse fasi di inizio.

Si tratta di scendere nelle viscere della terra per raggiungere e salvare dei minatori, rimasti intrappolati da un'eruzione vulcanica, evitando insidie, di vario genere, aprendosi passaggi e non perdendo la strada.

Per difendersi da ragni, insetti e serpenti vari abbiamo una pistola a carica illimitata. Per fare saltare le pareti, invece, dobbiamo usare delle bombe ad orologeria, che sono in numero limitato (7).

Per il controllo del nostro jetman possiamo usare il joystick oppure 4 tasti, che danno però un controllo molto grossolano del movimento.

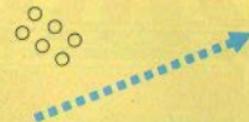
È possibile in qualsiasi momento sospendere («freeze») il gioco o riprenderlo dall'inizio premendo un tasto.

La grafica, come è pessima tradizione in casa Activision, è al limite dello squalore, i caratteri sono stilizzati e animati in maniera molto approssimativa e poco realistica. Non vi sono effetti sonori.

L'unica caratteristica degna di nota di questo programma consiste nel sistema di caricamento dal nastro. Infatti è stata riscritta la routine assembler che effettua materialmente il «load» per ottenere una maggiore velocità (circa il doppio del normale).

Questo sistema è stato adottato da praticamente tutte le software house per i programmi più recenti, dato che consente sia di ridurre l'attesa del giocatore, sia di mettere fuori gioco tutti i programmi copiatori (che non funzionano assolutamente), sia di rendere impossibile le copie illegali «nastro-a-nastro», dato che la qualità di riproduzione dev'essere perfetta. Garantiamo che il «forzare» questo tipo di protezione è molto impegnativo e richiede una grande esperienza: provare per credere.

- Grafica: stilizzata, approssimativa nell'animazione, monotona **5.**
- Comandi: essenziali ma poco efficienti, specie i tasti **5.**
- Giocabilità: nonostante i vari livelli ci si stanca presto **5.**



FINDERS KEEPERS
Mastertronic
Spectrum 48K.

Il re di Isbisima non ha idea di che cosa possa regalare alla figlia per il suo compleanno. Purtroppo tu sei proprio il cavaliere magico alle sue dipendenze e devi cercare qualcosa di molto speciale, per potere entrare a fare parte della Tavola del Poligono di corte.

Agisci nel castello di Spriteland, pieno di stanze ricche di oggetti e di nemici semoventi (che possono nutrirsi della tua energia vitale o rubare i tuoi oggetti). Alcuni oggetti possono combinarsi gli uni con gli altri per formarne di nuovi più preziosi.

Può anche scambiare o esaminare oggetti (tramite appositi tasti).

Finders Keepers è un gioco piuttosto originale, sofisticato nella trama e nella grafica multischermo, coloratissima. L'animazione è ad alto livello, tecnicamente perfetta. Notevoli le possibilità di esaminare o scambiare pezzi, che fanno assomigliare alcune fasi del gioco a un'avventura vera e propria.

Inoltre vi sono due possibili strategie di gioco: cercare di raccogliere il maggiore numero di oggetti possibile (punti) e fuggire quindi da Spriteland oppure cercare un regalo che soddisfi il re e sua figlia.

Bisogna dire anche che molto del fascino deriva dalla scarsità delle informazioni date dalle istruzioni accluse alla cassetta, che permette al giocatore di scoprire mentre gioca tutte le sorprese e le particolarità del gioco.

I comandi sono tanti: 9 tasti oppure 4 tipi di joystick più 4 tasti.

- Grafica: coloratissima, ben animata, varia, multischermo. **9**
- Comandi: comodi anche se numerosi; meglio le manopole. **7**
- Giocabilità: elevata, data la varietà e complessità di gioco. **8**



BEACH-HEAD
Us Gold/Ocean
Spectrum 48K.

Questo programma è stato convertito da un originale grande best-seller inglese per i computer Commodore 64. Per chi conoscesse quella versione, diciamo subito che è identica a questa per lo Spectrum, in cui la grafica è egualmente curata e tutti i particolari scrupolosamente ricopiati.

Si guida una flotta di navi all'attacco di una base navale nemica. Il gioco consta di 5 fasi diverse, anche se non sempre ben differenziate, visto che si cambiano solo 3 scenografie.

In una delle possibili due prime fasi (secondo la strategia scelta), è necessario battersi con le batterie contraeree contro i caccia nemici, che arrivano all'orizzonte per sganciare bombe sulle nostre navi, in una piacevole e appassionante grafica, con buona prospettiva.

L'alternativa è cercare di passare per una baia segreta, facendo uno slalom fra i siluri vaganti dei nemici; qui la grafica è meno avvincente e quasi banale.

Proseguendo si ritorna alla scena della battaglia aeronavale, per battersi con i cannoni contro le navi nemiche. Bisogna aggiustare il tiro, seguendo le indicazioni del radar («troppo lungo», «troppo corto»), fino ad affondare tutte e 4 le navi nemiche prima che loro affondino la nostra flotta. Praticamente bisogna trovare un numero tra 0 e 90 per tentativi, cosa abbastanza noiosa e frustrante.

Se si supera questa fase, dobbiamo fare un pezzo di navigazione molto pericoloso, dribblando vari tipi di ostacoli ad alta velocità, in grafica bidimensionale per imbarcattersi, finalmente, col cannone di protezione della base nemica.

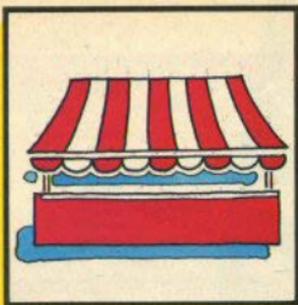
Questo cannone è tristemente famoso, perché non sbaglia mai un colpo...

Si può giocare da soli o in due, con i tasti o il joystick, su 5 livelli di difficoltà. È possibile ridefinire i tasti da usare.

Beach-Head è un insieme di cinque giochi: presi singolarmente sono abbastanza mediocri e poco originali, ma nel complesso formano qualcosa di abbastanza interessante e vario per gli appassionati dei giochi d'azione.

- Grafica: veramente buona solo in una delle fasi; colorata **6.**
- Comandi: molto comodi i tasti ridefinibili, joystick sconsigliato **8.**
- Giocabilità: buona grazie ai vari livelli ed alle varie fasi. **7.**





sinclair *reclame*

VENDO SCAMBIO HARDWARE

Vendo Joystick universale con manico tondo. Andrea Gherardi, via Q. Sella 12, 58100 Grosseto, tel. 490045.

Vendo sound board e light pen DK Tronics per Spectrum. Lorenzo Petersen, via Aurelia antica 391, 00165 Roma, 06/5239695.

Vendo Spectrum 48K + tastiera professionale DK Tronics + 150 video games + utilit  varie + 3 libri di istruzioni, L. 500.000. Paolo Sacchi, p.le Stazione 4, Pavia, 0382/38366.

Vendo Spectrum 16K, alimentatore e cavetti + «Alta scoperta dello ZX Spectrum», «77 programmi per Spectrum», «Programmi e applicazioni per lo ZX Spectrum», + cassette di software, L. 250.000, Giovanni Lombardi, Via Adige 24, 84091 Battipaglia, 0828/72160.

Vendo Spectrum 16K, cavetti e generatore, L. 250.000. Claudio Chiarinelli, via V. Emanuele 36, 20094 Corsico (MI), tel. 4473560 pasti.

Vendo Spectrum 48K quasi nuovo con garanzia, manuale, 150 programmi, L. 400.000 trattabili. Giovanni Quaranta, Via Duca Guglielmo 13, 84100 Salerno, tel. 396584.

Vendo Spectrum 48K, ZX printer, 2 rotoli carta, alimentatore 4A stab., 80 programmi su cassetta, floppy disk con 5 dischi pieni di programmi, L. 1.200.000, anche pezzi separati. Massimo Trevisan, via S. Iacopo 17, 56100 Pisa, 050/40792.

Vendo ZX 81 perfetto con cavet-

ti, alimentatore, manuale inglese + 4 libri in inglese, + programma "Chess", usato pochissimo L. 90.000. Francesco Bertani, via M. Bianchi 11, 20149 Milano, 02/466662 mattina.

Vendo Spectrum 48K, alimentatore, cavi, manuale, "The century computer programming course", "50 educational games for the Spectrum", cassetta Horizon e riviste, usato poco, L. 345.000 trattabili. Alessandro Briga, via S. Denis 185, 20099 Sesto S. Giovanni (MI), tel. 2485280 sera.

Vendo Spectrum 16K tre mesi, cavi, alimentatore, imballo e manuali originali, L. 275.000. Andrea Catini, via A. Poerio 8, 00152 Roma, 06/5817132 sera.

Vendo Zx interface + Zx microdrive come nuovi, imballo e manuali originali, L. 275.000. Stefano Pavanello, via Manin 10, 40026 Imola (BO), 0542/32779.

Vendo Spectrum 48K, ZX printer, mangianastri, manuale italiano, 20 giochi, in garanzia, L. 510.000. Enrico Romano, via Poggetto, 4, 16035 Rapallo (GE) 0135/54465.

Vendo Spectrum 48K 5 mesi, 2 manuali, alimentatore, cavi, interfaccia joystick programmabile, 150 programmi, L. 500.000 trattabili. Riccardo Abba, via Appennini 145, 20151 Milano 02/3532263 ore 19-22.

Vendo Spectrum 48K, manuale, libri inglesi, oltre 100 giochi e utilit . Paolo Faiaschi, via San Cristoforo 3, Ozzano nell'Emilia (BO), 051/797394 dopo le 19,30.

Vendo ZX printer + 4 rotoli carta metallizzata, L. 80.000. Paolo Ballocci, via San Gottardo 75, 20052 Monza, 039/367704.

Vendo Spectrum 48K issue 3 ottimo stato, L. 300.000. Claudio Lucchini, via del Carracci 7, 20149

Milano, 02/4690084.

Vendo Interfaccia joystick Kempston (usabile anche con la ZX interface), L. 40.000 trattabili; Interfaccia joystick Teknolek programmabile (guasta ma facilmente riparabile), L. 30.000; entrambe L. 80.000. Marco, Bologna, 051/333424.

Vendo Spectrum 48K con tastiera professionale, registratore Philips data recorder e + di 200 programmi, L. 400.000; senza registratore L. 340.000; regalo libri Jackson e Inglese. Andrea Pacelli, via Firenze 205, 65100 Pescara, 085/26390 ore 13,30-14,30.

Vendo Spectrum 48K issue 3 come nuovo, imballo originale, L. 350.000; con tastiera professionale Sandy L. 400.000; solo zona Siena. Michele Santillo, via E.S. Piccolomini 113, 53100 Siena, 0577/42925 ore 20,30.

Vendo Spectrum 48K, 300 programmi e box sonoro, L. 350.000 stampante Alphacom 32, L. 150.000; interfaccia joystick Kempston L. 15.000; registratore Philips D 6350 in garanzia, L. 150.000. Paolo Menini, p.za Europa 15, 44100 Ferrara, 0532/62436.

Vendo Spectrum 48K 1 anno ottimo stato, cavi, tasto reset all'alimentazione, manuale italiano, programmi vari, L. 500.000. Mauro Padovan, via Bellini 24, 47039 Savignano S.R. (FO), 0541/945790.

Vendo ZX 81 16K, 4 programmi originali, manuale inglese e italiano, libro "86 programmi per ZX 81", mascherine, tastiera, L. 100.000. Nicola Chelli, via Bowio 385, 65100 Pescara 085/72305.

Vendo Spectrum 48 K, manuale italiano, 10 giochi, interfaccia e joystick, varie, doppio regalo, nuovo, garanzia bianca, L. 450.000. Renzo Pedriali, via Codroipo 15, 41100 Modena, 059/305405 sera

Vendo Spectrum 48K, tastiera professionale con pad numerico, alimentatore, presa TV/monitor, manuale italiano, 4 volumi, 200 giochi e utilit , supergaranzia, L. 700.000 trattabili. Roberto Micheli, via Volvera 32 f, 10045 Piossasco (TO), 011/9066190 pasti

Vendo Spectrum 48K, manuale, libri e centinaia di programmi, ottime condizioni, 350.000. Roberto Ruscino, via Mons. Fr. Bruno 355, 98100 Messina, 090/49103

Vendo Spectrum 48K, 300 programmi inglesi, manuale italiano, cavi collegamento monitor, L. 340.000; stampante Seikooha GP 50S nuova già interfacciata, L. 250.000; entrambi, L. 570.000; disponibile prove. Maurizio Brugiolo, via Pio VII 138, 10127 Torino, 011/815827.

Vendo Spectrum 16K, alimentatore, cavi, manuale italiano, cassetta horizons, supergaranzia, buoni sconto per acquisto software, L. 250.000. Riccardo Bozoni, via Boeri 11, 20141 Milano, 02/8660921 dopo le 18,30.

Vendo Spectrum 48K, stampante Seikooha GP 250X, interfaccia Centronics RS232, tastiera professionale DK Tronics, interfaccia monitor, registratore specifico, 400 programmi, in ottime condizioni, L. 800.000. Giuliano Laudicci, via Piotti de Bianchi 4, 20129 Milano, 02/747985.

Vendo console Intellivision + modulo computer e 17 cassette anche separatamente, L. 500.000 trattabili. Emanuele Lupi, via Messina 46, 00198 Roma, 06/653138

Vendo Spectrum 80K, interfaccia 1, tastiera professionale (84 tasti) Kempston, interfaccia musicale a 3 voci, 300 programmi, libri e fotocopie, solo zona Torino. Roberto Testore, c. Monte Cucco 80, 10141 Torino, 011/728786.



Vendo ZX 81 espansione 32K Elettr. 32K Elettr. 2000, alimentatore maggiorato HL 203, cavi, libri tra cui "Guida allo ZX 81", "66 programmi", in inglese ecc., L. 260.000 + spese postali. Crete Dazio Porta, presso Malaguti, via Sturzo 18, 20091 Bresso (MI), 02/9140524 sera.

Vendo Spectrum 48K, ZX printer (nuova), 5 rotoli carta, innumerevoli programmi, L. 500.000. Maurizio Sorrentino, via Montefalcone 29/c, 19100 La Spezia, 0187/703862.

Vendo Spectrum 48K, registratore, interfaccia joystick programmabile, 200 programmi inglesi, L. 500.000; console Intellivision con 5 cassette. Mauro Alfano, via Vi settembre 36, 84100 Salerno, 089/321189

Vendo Spectrum 48K, manuale, libro su microdrive, interfaccia Joystick Kempston + joystick, 30 programmi, computer di 2 mesi in garanzia, L. 400.000. Sergio Macchione, via Galliano 15/9 Milano, 02/4592350 pasti.

Vendo Spectrum 48K, manuale, imballaggio ecc., circa 400 programmi inglesi, Filippo Montanelli, via Sirori 72, 22062 Barzano (CO), 039/955378

Vendo Spectrum 16k 4 mesi, alimentatore, cavi, manuale inglese e libro sul microdrive, 7 giochi, L. 295.000. Andrea Ricciuti, via Orsa minore 5, 20060 Cassina de' Pecchi (MI), tel. 9251458

Vendo Currah speech, zona Genova, L. 100.000 trattabili. Fabrizio Casalino, via De Nicolay 14 A/1, 16155 Pegli (GE) tel. 6857913.

Vendo Spectrum 16K 1 anno usato pochissimo, supergaranzia, imballo originale, alimentatore, cavi, manuale italiano, cassette "Horizon" e "Intruders" originali, L. 220.000. Franco di Molletta, via 20 settembre 24, 25070 Sabbio Chiese (BS) 0365/85151.

Vendo Spectrum 48k alimentatore, cavi, manuali inglese e italiano, un libro giochi + 60 giochi, imballo originale, L. 400.000. Massimo Bettle, via Gastaldi 18, 10023 Chieri (TO), 0119/423573

Vendo ZX Spectrum 48K, 3 mesi, alimentatore, manuale italiano, cavi, cassette di giochi e utilities, L. 270.000. Piero Iacovoni, via Marconi 9, 60044 Fabriano (AN), 0732/3002.

«programmare... Spectrum»; tutto il 1984 di «Papersoft» a ogni acquirente regalo rivista + il listato; zona Palermo e Trapani. Francesco Burgio, via Notarbartolo 49, 90100 Palermo, tel. 266096.

Trasferisco tutti i programmi su microdrive, anche quelli che iniziano sotto 29700. Lorenzo Petersen, via Aurelia Antica 391, 00185 Roma, 06/5239685.

Vendo Enciclopedia di elettronica e Informatica Jackson, L. 180.000. Giovanni Natale, viale Trieste 36, 93100 Caltanissetta, 0934/22775 pasti.

Trascrivo programmi. Francesco del Gaudio, via Quasimodo pal. Deranico, 87036 Rende (CS), tel. 872743.

Eseguo qualsiasi programma in Basic e L/M (eventualmente compilato); cerco interfaccia Kempston a buon prezzo o scambio con software. Umberio Ballestrazzi, via Tavoni 3, 41056 Vignola (MO), tel. 775048

Cerco Istruzioni in Italiano per master file, cerco bug per giochi vari. Paolo Migloranzi, via D.G. Trevisani 80, 37139 Verona.

Cerco stampanti Alphacom o Selkoha in cambio di Software per Spectrum. Francesco Golinielli, via Bellaria 28, 40139 Bologna, 051/545446.

Cerco penna ottica per Spectrum marca Tenkolos. Andrea Gherisani, via O. Sella 12, 58100 Grosseto, tel. 490045.

Vendo i seguenti libri: Jackson «Impariamo il Pascal» (lib. + cass.), «60 giochi per Spectrum», «77 programmi per Spectrum», «Programmazione ZX Spectrum», «Programmare immediatamente lo Spectrum» (lib. + cass.); McGraw Hill «ZX machine code assembler» (lib. + cass.); «Basic teoria ed esercizi», tutti a metà prezzo. Ernesto Bellucci, corso Isonzo 123, 44100 Ferrara, 0532/28759.

Cerco interfaccia per modem (e relativo software), scambio con 30 giochi su cassetta. Alessandro Carli via Rezzonico 41, 35100 Padova.

Compro software, testi, periferiche e programmi per Spectrum 48K. Pietro Ricciardi, via Principe di Napoli 7, 81052 Pignataro Magliore (CE).

Cerco «American Football» della Argus Press, Paolo Valentini, via Cavour 13, 36030 SelvaZZano (PD), 049/511657, ore 20,30-2.

Sinclair club Roma Talenti, per possessori Spectrum in tutta Italia. Rivolgerti a Massimo D'Ascenzo, via F. D'Uvidio 109, 00137 Roma, 06/8280043.

Compro programmi progettazione elettronica per Spectrum 48K o

scambio con utilities e giochi. Cerco stampante Selkoha GP 505 o Alphacom. Ilario Tariello, via Provinciale 3, 28072 Briona (NO), 0321/826222.

Cerco i programmi indicati nel software directory del n. 9 di SO: 3, 6, 10, 11, 28, 37, 42, 75, 78, 105, 185, 194, 206, 209, 244, 289, 294, 310, e numeri 147 e 234 con istruzioni. Giovanni Tagliavini, via Martiri di Cafelonia 14, 43100 Parma, 0521/569787.

Cerco Joystick Kempston per Spectrum con interfaccia, Giuseppe Celardo, via Soriano 1, 80011 Acera (NA), 081/8855854, pasti.

Cerco Spectrum anche 16K in cambio di software. Alfredo Santugli, via Carlotto 41, 00122 Ostia Lido (Roma), 06/5623424.

Cerco stampante per Spectrum, tastiera professionale e floppy disk (con informazioni da riviste in inglese). Pietro Mangiatordi, via Torti 113 rosso, 18143 Genova, 010/505283 (ore notturne).

Il club Byte di Cagliari è interessato a scambi e vendite con tutta Italia. Rivolgersi a: Carlo Atzeri, via de Giovanni 25, 09100 Cagliari, 070/305732.

Cerco stampante, o interfaccia 1 e 2, o microdrive, o qualsiasi interfaccia, offro programmi in cambio. Matteo Tossato, via Massingone 43, 36025 Noventa Vicentina (VI).

Desidero scambio informazioni con chi abbia lavorato con il programma «the Hobbit». Caterina Cavalli, via Dora Riparia 4, 20161 Milano, 02/9480822.

Cerco maniac di adventures per cassetta informazioni. Giuseppe Caccera, via Ferrarese 37, 81100 Caserta, 0823/324534.

È nato Sinclair club per Spectrum, iscrizione L. 10.000. Rivolgerti a Luigi Faberi, via Campagnola 3, 25079 Vobarno (BS), 0365/618388.

Sinclair club Foligno, presso Radio Delta, via C. Battisti 63, 06034 Foligno (PG), 0742/56607.

Compro espansione 16K per ZX 81, Fabio Morisio, via Biancamani 23, 20075 Lodi (MI).

Nuovo ZX club in Roma, l'iscrizione si paga in software. Rivolgersi a: Marco Di Silvestro, via dei Mamili 1, 00175 Roma.

VENDO SCAMBIO SOFTWARE

Vendono/scambiano

software (dove non specificato, si tratta di software per Spectrum).

Ernesto Bellucci, corso Isonzo 123, 44100 Ferrara, 0532/28759.

Maico Kelly, P.O. Box 2, 6961 Purasca (Ticino), Svizzera.

Possibilmente zona Varese. Sergio Zardo, via 4 novembre 24 A, 21040 Ubolido (VA), tel. 0639929.

Adriano Romiti, via Maroncelli 58, 58046 Marina di Grosseto (GR) 0564/35518.

Francesco Golinielli, via Bellaria 28, 40139 Bologna, 051/545446.

Zona Vicenza. Tel. 0444/546461 ore 16,30-19.

Paolo Migloranzi, via D.G. Trevisani 80 D, 37139 Verona.

Umberio Ballestrazzi, via Tavoni 3, 41056 Vignola (MO), tel. 775048.

Programma gestione spese familiari (fino 40 al mese), Istruzioni, compatibile microdrive, L. 30.000. Francesco Del Gaudio, via Quasimodo pal. Deranico, 87036 Rende (CS), tel. 872743.

Giovanni Natale, viale Trieste 36, 93100 Caltanissetta, 0934/22775 pasti.

Lorenzo Petersen, via Aurelia Antica 391, 00185 Roma, 06/5239685.

Francesco Burgio, via Notarbartolo 49, 90100 Palermo, tel. 266096.

Andrea Gherisani, via O. Sella 12, 58100 Grosseto, tel. 490045.

Maurizio Lattuada, via Panizoli 13, 20146 Milano, 02/427890.

Simone Fropini, via Casalpinho 20, 52100 Arezzo, tel. 353393.

Per Spectrum e Commodore 64. Raffaele Andrea, via E. De Marchi 79, 0141 Roma, 06/993162.

Piacenza e provincia. Edo Mars, via Gonzaga 22, 29100 Piacenza, 0523/71467.

Per Spectrum e ZX81. Mauro De Cupis, via del Giadoli 2, 00048 Nettuno (Roma).

Programmi su cartuccia. Via Lorenzoni Viani 21, 52142 Firenze, 055/711629.

Gianni Guattieri, via Slessa 21 A, 20135 Milano.

Paolo Della Capanna, via Fratti 784, 55049 Viareggio (LU), 0584/50390.

Per ZX81. Ferruccio Marelli, via S. Pietro 61, 12040 Govone (CN)

COMPRO CERCO VARIE

Vendo cerco «Inpu» aggiornato 40% del prezzo; libro + cassetta



Ultima versione del compilatore Pascal della HiSoft + manuali, e utilitès. Roberto Colotti, via Ticino 5, 35030 Sarmeola (PD), 049/631931.

Paolo Valentini, via Cavour 13, 35030 Caselle di Selvazone (PD), 049/631667.

Dino Ferliti, viale Scala greca 351, 96100 Siracusa, 0931/58665.

Alfredo Sanguigni, via E. Carlotto 41, 00122 Ostia Lido (Roma), 06/5623424.

Francesco Trasatti, via Collina 97, 63017 Porto San Giorgio (AP), 0734/48486.

Zona Treviso. Roberto Melchionno, via Ortigara 14, 31100 Treviso, tel. 52123 pasti.

Davide Di Dio, via Carlo Alberto 46, 80045 Pompei (NA), 081/9635055.

Per Sinclair QL. Roberto Ghezzi, via Volontari del sangue 202, 20099 Sesto San Giovanni, 02/2465511.

Per Sinclair QL. Luca Zammarchi, via Massetana 13, 58022 Follonica (GR), 0566/51511.

Gianluca Angelozzi, via Giullozzi 9, 62100 Macerata, 0733/32233.

Simone Cantini, via delle Palazzine 14, 50014 Fiesole (FI), 055/598758.

Luca Bonifazi, via C. Pascarella 3, 05100 Terni, tel. 407122.

Alessio Musti, via G. Armetelli 37, 00143 Roma, 06/5923320.

Zona Terni. Carlo Padronetti, via Stazione 41, 05100 Terni, tel. 411434.

Giuseppe Sciancalepore, via Paolo Emilio 50, 70059 Trani (BA), 0883/45682 pasti.

Daniele Ciuffardi, via Borachia 2, 19100 La Spezia, 0187/512537.

Edoardo Dell'Acqua, via A. Dionisi 81, 00151 Roma, 06/538635.

Marco Pagnini, via S. Biagio 212, 29011 Borgonovo V. T. (PC).

Toscana Spectrum club. Massimo Risaliti, via Leonardo da Vinci 13, 50042 Carmignano (FI).

Marco De Amicis, via L. E. Seiano 60, 00174 Roma, 06/7661837.

Programma oroscopo con dettagliate istruzioni e tabelle (effemeridi, case, tempo siderale, ore legali). Maurizio Balocchi, via A. Manzoni 31, 50015 Bagno a Ripoli (FI), 055/641668.

Fabrizio Mariano, via don Sturzo 7, 58100 Grosseto, 0564/492806 ore 19.30-20.30.

Per Sinclair QL, programmi, listati, articoli e utilitès da articoli e libri inglesi, Roberto Roberti, via Montello 8, 20094 Corsico (MI).

Enrico Busto, via campo sportivo, 10061 Cavour (TO).

Scambio programmi Spectrum con software MSX. Claudio Ziffero, via Tasso 40, 00185 Roma, 06/7314657.

Programma con manuale per sviluppare espressioni algebriche, L. 27000 contrassegni. Aldo Montaldo, via A. Zupi 19, 87100 Cosenza.

Bruno Ballarini, via Magenta 5, 10012 Bollengo (TO).

Emanuele Lupi, via Messina 46, 00196 Roma, 06/853136 pasti.

Mario Guarnaccia, via Mascagni 57, 95129 Catania, tel. 988034.

Alessandro Carli, via Rezzonico 41, 35100 Padova

Rosario Di Modica, via Castellardo 37, 97019 Vittoria (RG), 0932/963512 ore 14-15.

Ivano Parbuono, via A. Di Cambio 4, 37138 Verona, 045/568649.

Per Spectrum e Commodore. Andrea Voita, via M. Mazza 40, 16133 Genova, tel. 3997963.

Armando Acetulli, via Di Vittorio 8, 62100 Brindisi.

Alberto Profumo, via Romagnoli 28/2, 17100 Savona, 019/800797.

Giuseppe Cecere, via Ferrarecchio 37, 81100 Caserta, 0823/324534.

Giuseppe Ferrari, via Rivali 152, 42025 Cavriago (RE).

Adriano Biancotti, via Montegrappa 23, 10064 Pinerolo (TO), 0121/598363.

Claudia e Giulio Cavalli, via Dora Riparia 4, 20181 Milano, 02/648022.

I tasti

P

NORMALE

Statement PRINT. Un altro tasto dai molti usi; poichè non presenta particolari difficoltà, elenchiamo sinteticamente le sintassi più usate:

PRINT *numero*

PRINT "stringa"

PRINT *nome di variabile assegnata*

sono di significato evidente, indicando qualsiasi dei tre argomenti con [arg], si può avere:

PRINT AT x,y; [arg]

PRINT TAB x1; [arg]; TAB x2; [arg]

PRINT [arg1], [arg2], [arg3]... etc.

I vari attributi di stampa possono seguire una PRINT: in questo modo riguarderanno soltanto ciò che viene stampato con quella PRINT: il procedimento vale per INK, PAPER, BRIGHT, OVER, etc.

Esempio:

PRINT INK 3; PAPER 6; FLASH 1; "premi un tasto"

Non serve però solo per il video: si

può associare a PRINT un canale di comunicazione, indirizzando così il flusso di dati verso un dispositivo diverso. In realtà, PRINT ha sempre un canale associato: se date PRINT 2; "...", ottiene lo stesso risultato di un comando normale. Il 2 è infatti il canale automaticamente legato a "s" (screen); 0 e 1 sono per la parte bassa dello schermo, 3 è per la stampante; questi sono sempre aperti ed è preferibile non utilizzarli per altre direzioni, salvo casi particolari. Da 4 a 15 sono a disposizione per gli usi con l'Interface 1 (microdrives, RS232 e network).

SYMBOL SHIFT

Doppi apici (""). Fondamentale segno del basic, serve per identificare la stringa alfanumerica:

LET a\$ "abcdef"

PRINT "ghijkl"

LET q = CODE "mnopqr"

Entra poi nella sintassi di varie funzioni, in particolari in tutte quelle che riguardano l'Interface 1, per identificare i vari devices: "m" (microdrive), "n" (network), "t" e "b" (porta RS232); va usata per racchiudere i nomi da assegnare a

programmi (SAVE, LOAD, etc.), a cartridge (FORMAT), a files (OPEN, CLOSE).

Se volete che una stringa venga stampata con gli apici, non è sufficiente metterli doppi, come sembrerebbe logico: occorrono tripli:

PRINT "Questo è ""Sinclair Computer"""" darà in esecuzione
Questo è "Sinclair Computer".

CAPS SHIFT

"P" maiuscola.

MODO «E»

Comando TAB. Come AT, anche questa non vive senza PRINT (o eventualmente INPUT). Serve, come suggerisce il nome, a tabulare: indirizza la colonna di stampa sul video, senza alterare la linea corrente. Il range utile è ovviamente da 0 a 31: se date un valore inferiore alla posizione corrente, la stampa sarà alla riga successiva, se l'argomento di TAB è maggiore di 31, viene automaticamente ridotto a questo modulo.

SYMBOL S. in «E»

Simbolo di Copyright. Ha un uso esclusivamente grafico.



Indice analitico
di
SINCLAIR COMPUTER
(febbraio 84/gennaio 85)

Didattica

gli operatori logici 1,31
che c'è dietro il beep 1,45
canali di comunic. e flussi 4,6
programmazione in assembly 4,10
5,10 6,14 7,15 8,14 10,15
il quadrilettera 4,46
PEEK & POKE 5,6
conversioni in Assembly 5,32
variabili di sistema (Spectrum) 6,
46 7,19
variabili di sistema (ZX81) 8,13
messaggi di errore (ZX81) 8,49
computer music 8,45 9,60 10,19
le interruzioni 9,7
dallo ZX81 allo Spectrum 9,64
10,47
un tasto per volta 9,79 10,49
interrogiamo la tastiera 10,7
rilocabilità del linguaggio macchi-
na 10,43
messaggi di errore (spectrum)
10,45

Periferiche

microdrives anteprima 1,15
due stampanti, interfacce... (Alphacom
32 e Seikosha GP100) 2,6
i microdrives 3,6
il mistero della cartuccia 6,53
Seikosha GP50-S 7,17
MBI Tandem 7,52
floppy disk drive Sandy 8,16

Servizi

gli adventures I 8,8
"Kill wolf with sword" (adventures II)
9,19
carta, penna e bussola (advent. III)
10,13
matto allo spectrum 9,63

La ROM dello ZX81

1,35 2,37 3,36 4,35
5,29 6,48 7,55 8,54
9,70 10,53

Hardware

viaggio nello spectrum 1,8 2,10
3,11 4,30 5,8
espansione RAM 3,15
alimentazione spectrum 4,44
schede: lo Z80 6,12

il video-collegan. monitor 6,21
alimentaz. spectrum II 6,55
alimentaz. spectrum III 8,53
fatevi un monitor 9,66

Software ZX81

didattici:
analisi di un testo 1,27
seno 1,30
decimali 3,29
frazioni 3,32
dai romani agli arabi 4,20
piccola matematica 5,24

utility:
la locazione 16418 1,18
caratteri ingranditi 1,20
calendario 2,17
trasfer.pag.video 2,26
archivio 5,22
controllo della RAM 5,37
istogrammi 5,26
caratteri allungati 6,25,30
READ,DATA,RESTORE 7,43 10,42
la data della pasqua 8,26,37

giochi:
filetto 1,19
solitaire 1,23
piccoli universi 1,24
labirinto 1,26
la strada della dannazione 2,14
slalò 2,20
minimastermind 3,21
bioritmi 3,27
bomb 5,26
il computer dei destini incrociati
6,43,32
piramide 6,25,30
tombola 7,24,41
corsa di cavalli 8,44,39

Software Spectrum

didattici:
algebra 1,17
studio di funzione 2,22
autodraw 3,24
resistenze 3,25
travi 3,28
decimali 3,29
frazioni 3,32
dai romani agli arabi 4,20
binario 5,28
equazione il grado 6,26,31
print indirizzata 6,54,41
interpolazione Lagrange 7,21,28
sistemi lineari 7,23,31
calore disperso 7,26,34



intersezioni 9,59,53
analizz.audio 1 0,24,37

utilities

lettore di header 1,47 3,41 1 0,27
trasfer.pagina video 2,26/5,38
supermonitor 2,29
dump variabili 2,33
set caratt.corsivo 4,16
print doppia altezza 4,19
quadrilettera 4,46
verifica RAM 5,23
ingrandim.rilocabile 6,24,28
set caratteri 6,41
tape scanner 7,7
fill 7,22,29
 appendice 1 0,26,42
 protezione basic 7,22,31
 memory economizer 7,25,33
 delete 7,25,42
 tips & tricks 7,46,42
 DRAW assoluta 7,48,42
 set caratteri 7,63
 movimenti sullo schermo 8,21
 27,29
screen reductor 9,24,29
shell sort 9,25,31
orologio digitale 9,7,27

giochi

un coniglio nei pasticci 1,21
ufo qui,ufo la' 3,18
guerra aeronavale 3,22
 (errata corrige 5,39)
riflessi 3,30
un gioco per due 3,31
forza quattro 4,24
labirinto 3d 4,25
 (modificato, 8,27)
gather 4,28
hot dogs 5,13
ski 5,16
funghi per il folletto 5,19
oil panic 6,45,39
safari 6,45,48
carburante 7,26,36
tornado 8,24,34
virus attack 9,24,38
torre di hanoi 1 0,25,38
sfoglia la margherita 1 0,26,58

suono

tastiera ben temperata 4,23
un motivo per il fisco 6,26,31
speaker 7,44,38
spectrum tra le note/I 8,45,28
spectrum tra le note/II 9,60,54
i valori musicali 1 0,19,27

varie

totocalcio/I 4,14
totocalcio/II 5,13
aggiornam.totocalcio/I 6,22,27
aggiornam.totocalcio/II 8,63
lunazioni 4,21
traduttore 6,23,27
diesel o benzina 6,44,37
almanacco new-look 7,21,27

gioco dell'analista 7,45,39
confronto automobili 8,23,31
criptografie 8,26,38
arredare 9,26,32
conto corrente/I 1 0,23,32

Sinclair QL

anteprima 3,33
ling.macchina 8,53
prova 9,9

Recensioni giochi

SOFTWARE DIRECTORY 9,35
 (errata corrige 1 0,63)

Adventure directory 6,33

1,48

Jet Pac (Ultimate)
Xadom (Quicksilva)
Bubble Trouble (Arcade)
Tranz Am (Ultimate)

3,48

The Birds and the Bees (Bug-byte)
Arcadia (Imagine)
Chequered Flag (Psion)
Alchemist (Imagine)

5,46

Zip-zap (Imagine)
Cavern Fighter (Bug-byte)
Aqualane (Quicksilva)
Lunar Jetman (Ultimate)

6,58

Atic Atac (Ultimate)
Zoom (Imagine)
Bugaboo (Quicksilva)
Stonkers (Imagine)

7,58

Ant Attack (Quicksilva)
Transversion (Ocean)
Orc Attack (Creative Sparks)
Mosaico (JCE)

8,58

Mr.Wimpy (Ocean)
Eskimo Eddie (Ocean)
Kong (Ocean)
Hunchback (Ocean)

9,74

Psytron (Beyond Software)
Stop the express (Sinclair Res.)
Sabre Wulf (Ultimate)
Mugsy (Melbourne)

1 0,58

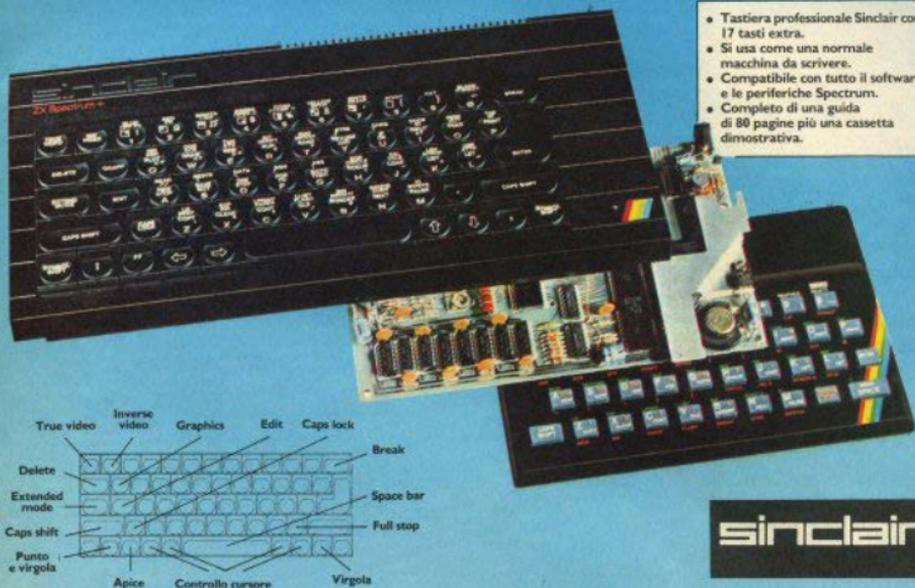
Krakatoa (Abbex)
3D Tank Duel (Realtime)
Tornado Low Level (Vortex/Ocean)
Android One (Vortex)

la prima cifra (car.espanso) indica il numero della rivista,
la seconda (normale) la pagina,
l'eventuale terza nel software (italic) la posiz. del listato.



Novità !! Lo Spectrum maggiorato !!!

Trasforma il tuo Spectrum in Spectrum +



- Tastiera professionale Sinclair con 17 tasti extra.
- Si usa come una normale macchina da scrivere.
- Compatibile con tutto il software e le periferiche Spectrum.
- Completo di una guida di 80 pagine più una cassetta dimostrativa.

Ecco una novità stimolante per i possessori di Spectrum: il **Kit originale Sinclair** che promuove lo Spectrum al grado superiore.

Non si richiede vasta esperienza. Basta saper saldare pochi fili.

Nel Kit sono contenute chiare istruzioni in italiano.

La tastiera dello Spectrum +

Le dimensioni dello "Spectrum +", sono 320x150x40. La tastiera, di tipo dattilografico, ha solidi tasti antiurto. Il suo uso è morbido e ideale per la scrittura al tocco, per il word processing, per i programmi di simulazione e i programmi avanzati.

Vi sono 58 tasti, di cui 17 rappresentano delle novità. I programmatori avranno la gradita sorpresa di trovare i tasti della punteggiatura e, a parte, i tasti "shift", per i grafici e gli "extended modes". Il tasto di reset consente di cancellare un programma dalla memoria del computer senza scollegare l'alimentazione.

Lo Spectrum maggiorato

Naturalmente il computer di grado superiore accetta tutte le periferiche del Sinclair System-Interface I, i Microdrives, eccetera, come pure tutto il software Spectrum.

I nuovi software e le nuove periferiche Spectrum saranno progettati tenendo conto dello Spectrum +, cosicché lo Spectrum accresciuto di grado avrà nuove capacità e potenziale nuovo per il futuro, oltre ad essere stilisticamente fantastico!!!



Il **Kit** contiene le istruzioni per l'assemblaggio, il manuale e la cassetta «demo didattica» in italiano.



In vendita presso i rivenditori specializzati



Utilizzate i tagliandi per abbonarvi, collaborare, chiedere o darci consigli, pubblicare un'inserzione per comprare, vendere, cercare contatti.

Registrate il mio abbonamento annuale a:

Sinclair Computer (L. 28.000) Computer + Sinclair Computer (L. 55.000)

Ho versato l'importo sul c/c postale n. 37952207

Accludo assegno non trasferibile n. _____ banca _____

intestato a SYSTEMS Editoriale, v.le Famagosta 75, 20142 Milano

Il mio computer è ZX81 Spectrum _____

possiedo SI NO

Nome _____

stampante

Via _____ n. _____

microdrive

CAP. [] [] [] [] [] città _____

joystick

Tel. _____

prov. _____

N.B. L'abbonamento è annuale (11 numeri) e decorre dal primo numero edito dopo il ricevimento della sottoscrizione.

Desidero collaborare a Sinclair Computer

COLLABORAZIONE

Invio il programma " _____ "

listato e registrato su cassetta, con un articolo di commento.

Garantisco che il software è originale e vi autorizzo a pubblicarlo.

Per il compenso scrivete mi al seguente indirizzo:

Nome _____

Via _____ n. _____

CAP. [] [] [] [] [] città _____

prov. _____

Tel. _____

N.B. Il materiale anche non pubblicato non viene restituito

HELP

Ritagliate e spedite in busta chiusa a: Sinclair Computer, viale Famagosta 75, 20142 Milano.

Nome _____

Via _____ n. _____

CAP. [] [] [] [] [] città _____

Tel. _____

prov. _____

Questo mese ho acquistato / provato i seguenti programmi e il valuto così (max tre titoli):

CLASSIFICA

TITOLO	Ottimo	Buono	Mediocre	Deludente

nome e indirizzo (facoltativo) _____

VENDO

COMPRO

INSERZIONI

Nome _____

Via _____ n. _____

CAP. [] [] [] [] [] città _____

Tel. _____

prov. _____



Avvertenze:

ABBONAMENTI: scrivete l'indirizzo completo

COLLABORAZIONE: il listato non è indispensabile, la cassetta sì.

HELP: non accludete francobolli, non rispondiamo privatamente

INSERZIONI: la rubrica è destinata agli scambi tra privati; la redazione si riserva il diritto di modificare o cestinare gli annunci palesemente speculativi.

INSERZIONI: la rubrica è destinata agli scambi tra privati; la redazione si riserva il diritto di modificare o cestinare gli annunci palesemente speculativi.

INSERZIONI: la rubrica è destinata agli scambi tra privati; la redazione si riserva il diritto di modificare o cestinare gli annunci palesemente speculativi.

ZX Spectrum Expansion System

L'alternativa della Sinclair ai floppy disc

Lo ZX Spectrum Expansion System contiene:

- **Uno ZX Microdrive** - Che amplia la possibilità dello ZX Spectrum in quei settori, come quelli della didattica e delle piccole applicazioni gestionali, dove è necessaria una veloce ricerca delle informazioni memorizzate su un supporto magnetico.
- **Una ZX Interface 1** - Indispensabile per il collegamento dello ZX Microdrive. Incorpora una interfaccia RS 232 e un sistema di collegamento in rete locale.
- **Quattro cartucce Microdrive comprendenti un programma di:**
 - Word processor «Tasword Secondo»
 - Masterfile filing system
 - Inventore di giochi
 - Le formiche giganti
- Un programma dimostrativo del Microdrive
- Documentazione per il collegamento, il funzionamento e altre descrizioni tecniche.
- Cavi di collegamento allo ZX NET che può collegare fino a 64 computer ZX Spectrum o QL.



In vendita presso
i rivenditori specializzati

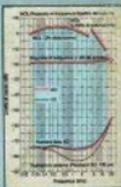


Le nuove Maxell

UDI

Vanta una tecnologia senza confronti nella sua categoria.

Infatti tutte le proprietà magnetiche del suo nastro sono esaltate al massimo grazie alla nuova particella magnetica "Ferricrystal" che, a differenza di quelle tradizionali, è



assolutamente non porosa. La UDI, quindi, offre un livello d'uscita migliorato su tutta la gamma di frequenze, specialmente nelle medie e basse; una più ampia gamma dinamica, ed ottime caratteristiche di lownoise. Adotta la meccanica P.A. (PHASE ACCURACY) per garantire la massima stabilità di svolgimento del nastro e per contenere la differenza di fase tra i canali stereo entro i 10°.

Può essere usata su qualsiasi tipo di registratore.

UD II

È la soluzione ideale per chi vuole una cassetta di categoria superiore ad un costo contenuto.

Utilizza un nastro di posizione "Chromo" (CrO₂)

prodotto con la stessa tecnologia dei nastri XL e XL-S Maxell.

Infatti il nastro della UD II è composto dalle collaudate particelle magnetiche "Fine Epitaxial", ulteriormente perfezionate. Di conseguenza è aumentata tutta la **gamma dinamica** del nastro, il livello d'uscita alle **medie ed alte** frequenze, mentre il livello di rumore di bias ed il rumore di modulazione sono ridotti praticamente a **zero**.

Il meccanismo di scorrimento P.A. e la perfezione dei gusci della cassetta assicurano un regolare svolgimento del nastro, mantenendolo sempre perpendicolare alla testina del registratore (differenza di fase entro i 10°).



maxell

È TUTTA UN'ALTRA MUSICA.

