

# Sinclair COMPUTER

mensile per gli utenti dei computer Sinclair

#17 Lire 3000  
ottobre 1985

A scuola  
col  
computer

Super Basic

Il linguaggio C

In edicola  
questo mese



L. 12.000

Sinclair  
COMPUTER

presentano

DAN PICUS e MIRANDVLA



**BANCA  
DATI**

VERSIONI PER  
**COMMODORE 64**  
&  
**SPECTRUM**

Ssystems

Il tuo  
**DATA BASE**  
personale

Suppl. a Sinclair Computer n. 16 - 1985 - Distr. MePe

- 04 - Sinclairmente vostro / la posta
- 06 - Recensioni libro
- 07 - Un computer per insegnante
- 08 - Il linguaggio C
- 10 - Il Database / II parte
- 16 - I canali dello Spectrum / II parte
- 18 - SIM Hi-Fi Ives
- 19 - Software: Window
- 20 - : Astrolabio
- 21 - : Simon / Slot-machine
- 22 - : Chitarra
- 23/39 - I listati
- 40 - Un po' di PASCAL
- 43 - Recensioni videogames
- 46 - Software: Virgola mobile / Scroll orizzontale
- 47 - spazio QL / Il sistema operativo
- 48 - / Hard-copy del video
- 50 - / Superbasic
- 50 - / JS: la nuova ROM

**SINCLAIR  
COMPUTER**

**REDAZIONE**  
Mauro Soliviani, Fabio Bero,  
Marco De Martino

**SEGRETERIA DI REDAZIONE**  
Mauro Ceccaroli, Piera Perin

**COLLABORATORI**  
Paolo Beneventi, Marco Bertani, Carlo  
Bolchini, Damiano Bolla, Giuliano Bo-  
sch, Luigi Callegari, Giampaolo Carri, Vale-  
rio Cipolla, Paolo Dray, Fabrice Ferraro,  
Monica Fumagalli, Guido Grassi, Giovan-  
ni Mellina, Marco Mussini, Roberto Pre-  
tera, Renzo Zonin.

**GRAFICA E IMPAGINAZIONE**  
Miry Benvenuto, Arturo Ciaglia, Paolo  
Vertuccio.

**DIFFUSIONE E ABBONAMENTI**  
Marina Vantini

**DIREZIONE, REDAZIONE**  
Viale Farnagosta 75 - 20142 Milano -  
Tel. (02) 8467348/9/40

**PUBBLICITÀ**  
Milano: Mirco Croca (coordinatore), Mi-  
chela Prandini, Giorgio Ruffoni, Claudio  
Tidone, Villa Claudio  
Segreteria: Lilliana Degliori  
V.le Farnagosta 75, 20142 Milano - tel. (02)  
8467348/9/40  
Roma: Spazio nuovo via P. Foscarini 70,  
00139 Roma tel. (06) 8109679

**FOTOCOPOSIZIONE**  
Fotocomposizione LM (Brescia)

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

**DISTRIBUZIONE**  
Messaggerie Periodici S.p.A.  
via G. Carcano 32, Milano  
Spedizioni in abb. Post. GR. 10/70

**SYSTEMS EDITORIALE s.r.l.**  
(Registro Nazionale Stampa  
n. 01500 vol. 15 fg. 793)  
Direttore responsabile:  
Agostina Ronchetti  
Autoriz. Trib. di Mi n. 255/12.11.1983

Una copia L. 3.000 (Arretrati L. 6.000  
Abbonamento annuo (11 numeri) L.  
26.000 (estero il doppio). I versamenti e le  
richieste di arretrati vanno indirizzati a:  
Sinclair Computer, V.le Farnagosta 75,  
20142 Milano, mediante emissione di as-  
segno bancario o versamento sul c/c po-  
stale n. 37952207 intestato a SYSTEMS  
editoriale. Per i cambi di indirizzo indica-  
re, unitamente al nuovo, anche l'indirizzo  
precedente, allegando L. 500 in franco-  
bolli.

Sinclair ZX81, ZX Spectrum, ZX Micro-  
ve, QL sono marchi registrati della Sin-  
clair Research Ltd.



LA POSTA LA POSTA LA POSTA LA

STA LA POSTA LA POSTA LA POS



# sinclair *amente vostro*

**Esiste un modem da collegare allo Spectrum? Se sì, qual'è, e come si collega? (vari lettori)**

I modem, sia integrati, cioè allacciati direttamente alla rete telefonica, sia ad accoppiamento acustico, che usando la cornetta di un normale telefono, si collegano al computer attraverso una porta seriale (RS232-C). Perciò qualsiasi modem è in linea di massima collegabile allo Spectrum, perchè questo sia dotato di Interfaccia 1. Altra cosa è però il funzionamento: il software necessario, per poter dialogare senza problemi con altri computer, deve essere realizzato con molta cura. Stiamo mettendo a punto un simile programma, sulla spinta delle numerose richieste, e lo pubblicheremo prossimamente.

**Le stampanti Seiksha GP100-VC e GP500-VC sono collegabili allo Spectrum? (A. Taricco Bergamo)**

No: le Seiksha serie VC sono interfacciate per Commodore, il cui controllo per la stampa non è standard. Si tratta peraltro di stampanti con prestazioni identiche alle GP100 e 500 normali.

## Monitor

Continuano a giungere in redazione domande sul monitor che si possono usare e come devono essere collegati allo Spectrum o al QL. Ripetiamo ancora una volta

(speriamo sia l'ultima) che allo Spectrum non può essere collegato in nessun modo un monitor che richieda segnale RGB, a colori separati; sono invece facilmente utilizzabili i monitor a colori con segnale composito, e ovviamente tutti i monocromatici. Qualsiasi monitor, RGB compresi, può essere usato con il QL, anche se per poter sfruttare la massima risoluzione occorre un apparecchio appositamente tarato.



Quanto al modo di collegare video e computer, vi rinviamo all'articolo apparso sullo scorso numero; in generale, l'unica cosa che vi serve è la piedinatura del connettore del monitor.

Alcuni lettori lamentano la non reperibilità del software recensito nelle rubriche giochi: effettivamente in alcuni casi sono state presentate "primizie", che può essere difficile reperire, soprattutto fuori dalla grossa città: i distributori non hanno mai brillato per sollecitudine e capillarità. Vi assicuriamo comunque che i giochi esistono tutti: le schermate riprodotte sono sempre fotografie originali eseguite da noi.

**Esiste un comando diretto per programmare in assembler? Quali comandi bisogna usare per sostitu-**

**re ai tasti l'interfaccia Kempston? Come funziona la cassetta pulisci testine? (M. Monteforti - Pontedera)**

L'assembler è un linguaggio e, come il basic o il Pascal, richiede un compilatore, che funga da "interfaccia" con la CPU. Il fatto che il basic nel microcomputer sia residente in ROM non cambia i termini della questione (i grossi computer non hanno mai linguaggi residenti). Naturalmente, nulla vieta di fare a meno del programma compilatore, scrivere l'assembler sulla carta e caricare direttamente i codici nel computer: se vi piace tanto fare a mano anche il debug...

Le interfacce joystick in generale usano l'istruzione IN, presente sulla tastiera dello Spectrum per la gestione degli input: cfr. la rubrica "Un tasto per volta", pubblicata in più parti sulla rivista, in cui abbiamo analizzato tutte le funzioni disponibili.

Le cassette pulisci-testine più comuni sono sconsigliabili, in quanto abrasive: riducono di molto la durata delle testine del registratore. Esistono prodotti più sofisticati e decisamente più affidabili, dedicati all'Hi-Fi: il sistema classico e "povero" più in uso e costituito da uno stuzzicadenti, un poco di cotone e l'apposito liquido detergente.



Supponendo di voler mantenere in memoria due programmi diversi contemporaneamente, uno sotto e uno sopra RAMTOP, quali variabili di stato è necessario modificare per "commutare" dall'uno all'altro? (L. Ghisleni - C. Magnago VA)

Non è possibile tenere in memoria due programmi basic in aree di verde della memoria nel modo che ipotizzi: sopra RAMTOP si può collocare soltanto linguaggio macchina. Questo perché tra la fine dell'area basic/variabili e RAMTOP esiste un'area di RAM in cui il processore colloca stack di lavoro a suo uso e consumo, non trasparenti all'utente.

**Come si usa il word-processor Tasword con le stampanti Seikosha? Come si usa lo Spectrum Writer con le Epson e Contabill? hard-copy con printer con Epson. Abbinamento Sp. Writer e interf. Kempston. (vari)**

Il word-processor TASWORD II è previsto per l'uso con stampanti Epson e compatibili, e non funziona con le Seikosha. Al contrario, la sua versione italianizzata Spectrum Writer prevede l'impiego delle Seikosha, e non funziona con le Epson e simili.

Allo stesso modo, le interfacce e il software per eseguire listati, stampe diverse e hard-copy è diverso per le due serie di stampanti: occorre controllare al momento dell'acquisto che il prodotto sia quello compatibile con la stampante (o viceversa): in alcuni casi il programma è configurabile per diversi tipi di stampanti, modificando alcuni parametri, in altri è fisso e funziona con un solo modello.

A quasi un anno di distanza riceviamo ancora richieste di invio di software dal catalogo pubblicato sul n. 09: ripetiamo che noi non vediamo quei programmi, li abbiamo solo catalogati per offrirvi un elenco ragionato di ciò che allora (ormai è totalmente superato) abbiamo trovato in circolazione, precisando (ancora a suo tempo) che non tutti i programmi si trovano in Italia.

È naturale che il trasformatore scaldi più del computer? È dannoso tenerli a lungo accesi, e tenere collegati i cavi del registratore? Il programma Music Colpiter del n. 11 (febbraio) va in errore quando inserisce le note. Nel listati appare il simbolo , non presente sulla tastiera dello Spectrum. (E. Baldo - Monza)

Non dovrebbero scaldare molto né il computer né l'alimentatore (il "molto" è un parametro soggettivo, direte voi: insomma, un uovo versato sulla tastiera non dovrebbe friggere...), ma certo è meno rischioso, anche economicamente, che scaldi un po' tanto il secondo piuttosto che il primo. Lo Spectrum può restare acceso per molte ore senza problemi; i cavi possono rimanere inseriti in permanenza (a parte togliere EAR durante il SAVE); Music Compiler (uscito in due parti sul n. 11 e 12) funziona regolarmente, ma occorre seguire attentamente le istruzioni; il segno (accento circonflesso) è il codice ASCII 94, che indica l'elevamento a potenza (ma quante volte dobbiamo ripeterlo ancora?), che lo Spectrum scrive sul video come una freccia rivolta in su (non molto diversa, ci sembra). Purtroppo per voi, tutte le stampanti in circolazione seguono lo standard ASCII e stampano il simbolo in questo modo.



**Tenendo la radio presso il computer, quest'ultimo provoca disturbi all'ascolto: si può rimediare? (P. Passavanti - Alba Cn)**

No: la circuiteria del computer, con gli oscillatori al quarzo, genera radiofrequenze che "sporcano" irrimediabilmente l'ambiente circostante, a meno di tenere la macchina entro contenitori schermanti. L'emissione è però di potenza molto limitata, basta allontanare un po' la radio.

Ho saputo dell'esistenza di schede per la ricezione del televideo. È possibile una simile scheda per lo Spectrum? Ne pubblicherete il progetto? (A. Ruggiero - Firenze)

la scheda per la ricezione del televideo non è così semplice come sostieni nella lettera, se si desidera un risultato almeno decente. Non abbiamo notizia, per ora, della commercializzazione di schede apposte per lo Spectrum, ma sappiamo di privati che le hanno costruite in proprio.

**Ho tentato di collegare il mio QL a un monitor RGB tramite presa SCART, ma la conseguenza è stata la perdita permanente del colore (anche col normale TV). Che cosa può essere successo? Pubblicheremo lo schema hardware del QL? (L. Rimassa - Sapierdarena GE)**

Non hai precisato come è stato collegato il QL. Non abbiamo in programma di pubblicare lo schema hardware.

**Qual'è il miglior assemblatore per lo Spectrum? Si può avere il disassemblato della ROM ombra? (G. Tagliavini - Parma)**

Il nostro personale e opinabile parere è che il migliore resti sempre quello della Hi-Soft, ma ve ne sono altri di ottimi.

Il disassemblato della ROM ombra dell'Interfaccia 1 è disponibile in inglese, pubblicato sempre dalla Melbourne, editrice anche del libro sul sistema operativo dello Spectrum; l'autore è Gianluca Carri, un nome già noto ai lettori di SC.

**Perché nei listati dei programmi in linguaggio macchina usate sempre i codici esadecimale, anziché i più comodi e semplici decimali? (D. Leoni - Firenze)**

I listati in linguaggio macchina vengono riprodotti come il stampa il programma disassemblatore. Si preferisce l'uso della numerazione esadecimale per vari motivi: si inconfondono più ordinatamente, essendo sempre di due cifre; risultano più facili la conversazione in binario, la separazione in byte alto e byte basso, e altre operazioni frequenti durante la programmazione e soprattutto nel debug dei programmi.





# Un computer per insegnante

di Pav

A una festa di fine anno, in una scuola media nell'hinterland di una città industriale del Nord, festina con canti, balli, abbuffate, libri e oggettini in vendita per raccogliere qualche lira a sostegno del magro bilancio, i partecipanti si affollavano davanti alla porta di un'aula. Dentro, un manipolo di zingarelli elettronici, per poche lire, faceva l'oroscopo con il computer. Per la cronaca, un C64, di proprietà della scuola.

Dicono i nuovi programmi per la scuola elementare: "La scuola attua i suoi compiti secondo un disegno preordinato, cioè seguendo le linee di un programma". Giocando sul doppio senso, viene spontaneo domandarsi in che linguaggio sia scritto il programma di cui seguir le linee. Basic, Logo, o che altro?

Le direttive prevedono che anche nella scuola inferiore, a fianco dell'inglese, entri l'informatica. Ma come e quando nessuno ancora è in grado di dirlo: dal prossimo anno scolastico, dicono i proclami ministeriali, e già alla maggioranza della popolazione coinvolta vengono i brividi. Ad alcuni di piacere, al più di smarrimento.

Nel frattempo, ogni scusa è buona per produrre carta stampata. Così, cercando documentazioni, ci siamo forzatamente resi conto del caos che regna sull'argomento: un mercato inflazionato, sia di hardware che di software, in cui nemmeno i ben informati possono essere certi di compiere le scelte migliori; le disposizioni ministeriali vagheggiano di una non meglio precisata introduzione dell'informatica nella scuola elementare e superiore, ma sembra che si

siano dimenticati la media; nessuno o quasi ha ancora capito se si vuole un'informatica come materia a sé, o una "metadisciplina", che entri orizzontalmente in tutto il settore tecnico-scientifico, e forse anche in quello umanistico.

I mass-media, dal canto loro, producono le consuete valanghe di parole: durante la scorsa primavera a parte dell'estate più o meno tutti i giornali periodici, specializzati e non, sono intervenuti sul tema informatica/didattica, pendolando con oscillazioni di pari ampiezza dall'accettazione acritica al rifiuto aprioristico, con tutte le sfumature possibili.

Nel marasma di software prodotto, imitato, copiato, c'è una quantità di cassette o dischi che si presenta come possibile strumento "didattico". Giochi più o meno ben vestiti in cui bisogna sparare alle desinenze sbagliate, giustamente "aliene" (è solo un esempio, vorrei dare un'idea del metodo). Ci sono buoni programmi per le materie tecniche, come aritmetica o chimica: guidano al disegno di una proiezione o al bilanciamento di una reazione acido-base. Cose che si potrebbero fare, al livello appunto didattico, altrettanto bene dal vero, senza il computer, senza il rischio di alleneare l'allievo dalla vita reale e dall'insegnante. Non mi sembra una grande trovata, come criterio didattico.

La smania di mettersi in mostra dei presidi d'assalto non va quasi mai di pari passo con un'adeguata e preventiva informazione: così accade che si spendano alcuni milioni per far acquistare al-



la scuola un personal di livello professionale, quando con la stessa cifra si sarebbero potute avere dieci o più macchine dalle prestazioni limitate, ma con quali un'intera classe potrebbe lavorare senza doversi contendere l'unica tastiera.

Non si tratta di illusioni: nel II Convegno internazionale sull'informatica nella scuola, tenuto a Castiglione (Livorno) nel marzo scorso, è emerso che un'altissima percentuale dei computer acquistati dalle scuole si trova inutilizzata (praticamente in cantina) o sottoutilizzata, o convertita ad applicazioni completamente diverse da quelle previste.

Esistono anche iniziative positive, naturalmente, spesso affidate alla volontà e creatività di un singolo.

Le scuole "sperimentali" conducono operazioni d'avanguardia, restando sostanzialmente isolate, proprio perché troppo d'avanguardia, e quindi non ripetibili e generalizzabili.

Ben dieci anni fa un insegnante di una scuola elementare di Trieste introdusse la programmazione nel suo piano di lavoro, utilizzando quello che allora era disponibile, cioè calcolatrici programabili.

Il progetto IRIS (Iniziativa e Ricerche per l'Informatica nella Scuola), quadriennale, è un esperimento condotto su 22 scuole di diverso grado sui metodi di insegnamento della nuova materia.

Le associazioni di categoria promuovono in po' ovunque corsi di "alfabetizzazione" informatica per gli insegnanti, che lamentano un generale disorientamento di fronte alla fuga in avanti degli allievi, che padroneggiano le "macchine infernali" molto meglio di loro.

Spesso poi questi corsi, tenuti a loro volta da personale tecnicamente qualificato ma didatticamente inetto, si risolvono in piatti e noiosi corsi di programmazione in basic, quando non in un qualsiasi dialetto basic, condizionato dallo sponsor che offre la macchina.

Non a caso il CNR (Centro Nazionale Ricerche) coordina sperimentazioni sul Logo e, in generale, sulla programmazione strutturata: dal punto di vista pedagogico il basic viene considerato un pessimo strumento, essendo un linguaggio non strutturato che "fa troppe cose" e consente soluzioni contorte, prive di blocchi logici: i programmi che gli americani chiamano *spaghetti-like*.

Purtroppo il mercato ha saldamente ancorato la maggior parte del pubblico al basic: "la rivoluzione informatica è appena agli inizi, e già sta generando un suo conservatorismo" ha detto Seymour Papert, il progettista del Logo.

# Il linguaggio C

di Pav

Da qualche tempo si sente parlare con insistenza di un nuovo linguaggio, il "C": nuovo per il settore dei microcomputer, poiché su grossi sistemi esiste già da vari anni (il suo successo è iniziato intorno al 1978); il "C" deve la sua fortuna al fatto che la maggior parte del sistema operativo UNIX (marchio registrato dei Laboratori Bell), lo standard che si è imposto sui grandi calcolatori, è stata scritta con esso.

Con l'aumento della velocità e della disponibilità di memoria sui personal computer, la distinzione tra "grandi" e "piccoli" diventa sempre più aleatoria: una barriera significativa è stata abbattuta dalla Hewlett-Packard, con la presentazione di un portatile che implementa l'UNIX come sistema operativo.

L'innovazione ha scosso anche la stampa più restia ai facili entusiasmi, tanto che qualcuno, un pò frettolosamente, è giunto a chiedersi se i grossi mainframe non siano già destinati a scomparire come i dinosauri.

Tutto questo che c'entra con il QL o con il simpatico ma inesorabilmente piccolo Spectrum? C'entra, perché per entrambi sono disponibili dei serissimi compilatori "C": il QC della GST Computer e (annunciato) un "C" della Lattice, commercializzato direttamente dalla Sinclair, per il QL; il "C" della Hi-Soft, la casa ben nota i suoi ottimi assembler/monitor, Pascal, etc., per lo Spectrum.

Inoltre, con il QL al pieno della sua potenza per quanto riguarda l'hardware, cioè con espansioni

RAM e degni supporti di massa (floppy ad alta capacità o hard-disk), non è affatto improbabile che presto si possa avere l'UNIX a disposizione anche su questa macchina.

## La portabilità

Tutti i libri e gli articoli che leggerete sul C esordiranno mettendo l'accento sulla trasportabilità come una delle prerogative più importanti del linguaggio: in effetti, abbiamo detto che il C ha avuto fortuna perché usato per scrivere l'UNIX, ma è anche vero il contrario - che l'UNIX si è imposto come nuovo standard perché scritto in C.

Il C non è un linguaggio a livello basso come un assembler, strettamente legato alla CPU: è ancora abbastanza leggibile, molte parole-chiave sono già familiari a chi conosce il basic o il Pascal, e risulta abbastanza facile da imparare.

Non è nemmeno a livello "alto" come il basic, il che gli ha risparmiato di differenziarsi in dialetti legati alle singole macchine, garantendo la quasi completa portabilità e l'indipendenza dei sistemi operativi e dalle macchine.

È molto più efficiente di un basic, e abbastanza poco meno efficiente di un assembler da consentire di scrivere un sistema operativo quasi tutto in C - appunto l'UNIX. Tanto che uno degli autori confessa nell'introduzione del manuale di non conoscere affatto l'assembler del PDP-11, la macchina su cui è stata scritta la prima versione.





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



**CONCESSIONARI MEZZI  
RADIOTELEVISIVI**

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



# Introduzione al



DATA  
BASE

di Mariangela Guardione

*Seconda parte della  
teoria del database: con la  
nostra cassetta sempre in edicola*

## Padri e figli

Questi vincoli possono essere così descritti:

- ogni padre può avere più figli;
- ogni figlio appartiene ad un solo padre;
- sono possibili collegamenti tra livelli contigui, nel senso che non risulta possibile eseguire un collegamento figlio-

nonno senza passare attraverso il padre;

- non sono possibili collegamenti tra livelli uguali; questo significa che due fratelli non sono in comunicazione fra loro;
- vi è un solo possibile "verso" di collegamento che è quello della direzione radice-foglie dell'albero; questo implica che non risulta possibile "ciclare" nella struttura, in quanto un elemento non può

essere contemporaneamente figlio e padre di un altro;

- un elemento appartenente alla struttura non può essere figlio o padre di se stesso.

Tutte queste regole rivestono un ruolo molto importante e devono essere rispettate sia durante la progettazione che la realizzazione e nell'utilizzo di un



data-base con struttura gerarchica, la cui rappresentazione è data in figura 3.

In un data-base così strutturato l'unità logica completa di informazione è un intero albero, in quanto contiene tutte le informazioni relative a una determinata situazione; queste sono costituite da sottoinsiemi di informazioni che dipendono da altri sottoinsiemi e le dipendenze esistenti fra essi sono regolate dai vincoli precedentemente descritti.

Lo schema logico di una struttura ad albero è costituito dall'insieme delle entità presenti nella struttura dei dati e dei collegamenti logici che esistono fra di esse.

Per meglio spiegare la struttura gerarchica, si riporta questo esempio relativo all'organizzazione e alla gestione di una biblioteca.

Tramite un data-base gerarchico risulta infatti ben rappresentabile la dipendenza Libro <-> Collocazione, in quanto ad ogni libro (padre) sono associate più collocazioni (figli); al contrario ogni collocazione appartiene a un ben determinato libro. Si rispettano così le regole generali precedentemente descritte e rappresentate graficamente in figura 4.

Nel caso in esame il data-base risulta essere a due soli livelli, in quanto a ogni libro è possibile associare tutte le informazioni che lo riguardano collocandolo quindi, nella struttura, come entità costituita da insiemi di dati elementari che possono ad esempio essere:

- titolo;
- editore;
- luogo di edizione;
- data prima edizione;
- prezzo.

Anche la collocazione risulta essere un'entità costituita da dati elementari come:

- sala;
- scaffale;
- ripiano;
- posizione.

chiamati comunemente "chiave di collocazione dei libri".

In un data-base gerarchico, come quello dell'esempio precedente, deve essere consentito scorrere l'albero partendo dalla radice, per inoltrarsi tra i rami specificando opportune condizioni e giungere a ottenere le informazioni volute. Inoltre deve esistere la possibilità di accedere al data-base mediante le chiavi che rappresentano uno qualsiasi dei dati elencati a qualsiasi livello essi si trovino. Da questo si deduce che non risulta obbligatorio visitare un albero dal nodo radice. Il legame esistente fra padre e figlio viene realizzato introducendo una "catena di puntatori", in quanto ogni padre contiene fra i suoi dati elementari un puntatore al primo dei figli che, a sua volta, ha un puntatore al fratello successivo e così via.

L'ultimo dei figli contiene un puntatore che indirizza al padre, chiudendo quindi la catena come viene mostrato in figura 5.

In un data-base gerarchico, quindi, deve essere possibile eseguire le seguenti operazioni:

- visitare l'albero secondo la sua naturale struttura gerarchica;
- accedere a un elemento dell'albero utilizzando una chiave di ricerca che eviti l'accesso a tutti i padri di quell'elemento;
- accedere ai figli di un determinato padre attraverso una catena di puntatori.

In un contesto di questo tipo è possibile eseguire le seguenti operazioni:

- utilizzando una chiave di ricerca, leggere un elemento della struttura gerarchica;
- accedere in lettura a un elemento della struttura che sia legato a uno precedentemente letto;
- leggere in catena il fratello del precedente;
- inserire un nuovo dato e collegarlo con i propri parenti;
- cancellare un elemento e quindi scollegarlo da tutti i suoi parenti;
- sostituire il contenuto di un elemento;
- inserire un intero albero;
- cancellare un intero albero.

E' possibile avere in realtà situazioni che, non essendo di natura gerarchica, non possono essere descritte con una struttura di questo tipo.

Infatti nel caso in cui vi siano due entità C e D legate da una relazione di tipo M:N e tra loro non sia possibile stabilire che C è il padre di D e viceversa, in quanto sono dello stesso livello e quindi legate fra loro, sarà necessario assumere che uno di essi sia effettivamente il padre e questo porterà ad una duplicazione di informazione nel data-base.

## Modello reticolare

Questo tipo di struttura usata per creare un data-base si fonda sostanzialmente su concetti basilari quali:

- record e tipo di record;
- set.

Il tipo-record rappresenta l'unità logica di informazione in cui sono presenti più dati elementari il cui insieme costituisce un'entità.

Se questo tipo-record risulta essere accessibile direttamente, l'insieme di uno o più dati elementari viene definita come "chiave del record" e in questo caso l'insieme dei records è visto come un normale file. Al contrario il set rappresenta il collegamento fra due tipi-record e risulta avere una direzione preferenziale che va dal tipo-record owner a quello member.

Il record può essere selezionato attraverso la propria chiave, che viene detta "calc" o mediante un set di cui il record è member; tutto questo non esclude però che il record abbia la possibilità di essere indirizzato in entrambi i sensi.

Anche nel caso di una struttura reticolare esistono dei vincoli da rispettare nella costruzione del data-base, che sono:

- il tipo-record owner può avere più "members";
- il tipo-record member può avere più "owners";
- il tipo-records non può essere contem-



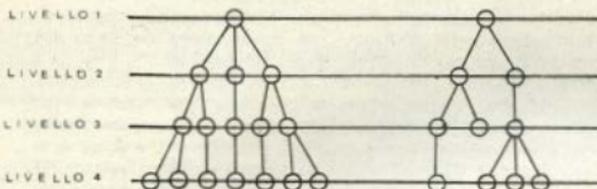


Fig. 3

poraneamente "owner" e "member" di un altro;

- il tipo-record non può essere contemporaneamente "member" o "owner" di se stesso;

- il set può collegare più tipi-record che risultano legati ad uno stesso tipo-record "owner".

Per spiegare in maniera più chiara cosa comportano questi tipi di vincoli sulla struttura logica del data-base, si applicano le seguenti regole alle tre dipendenze esistenti tra due insiemi di dati:

● **dipendenza 1:1**; in questo caso si possono verificare le due possibilità:

a) in cui vi è un solo tipo-record che contiene gli insiemi dei dati come indicato nelle figure 6 e 6a;

b) in questo caso, con due tipi-record collegati fra loro mediante un set, si ha un insieme di dati owner come indicato in figura 7;

● **dipendenza 1:N**; in questa situazione, a seconda dell'esistenza o meno di

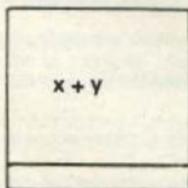


Fig. 4

una chiave del record member, questo sarà indirizzato "calc" o set come in figura 8;

● **dipendenza M:N**; in questa situazione, poiché non è possibile che un tipo-record sia contemporaneamente member e owner di un altro, è necessario definire un tipo-record quale legame di connessione fra i due come in figura 9.

A illustrazione del modello reticolare si riporta il seguente esempio, sempre relativo all'organizzazione e alla gestione di una biblioteca.

Volendo memorizzare in un data-base di tipo reticolare le informazioni riguardanti i prestiti dei libri effettuati dalla biblioteca, si dovranno definire un tipo record denominato "lettore" e uno denominato "libro", legati da una corrispondenza M:N e memorizzanti rispettivamente i dati relativi al lettore e al libro.

La corrispondenza utilizzata è imposta dal fatto che lo stesso libro può essere prestato a più lettori che potranno avere in prestito più libri. Per meglio definire quale libro di un gruppo con identico titolo è stato prestato ad un determinato lettore, è necessario stabilire un legame di tipo 1:N, tramite la collocazione del libro specifico nella biblioteca.

### Modello relazionale

Questo modello rappresenta una schematizzazione teorica per la strutturazione logica dei dati proposta da E.F. Codd nel 1970 e si basa sull'unico concetto di relazione utilizzato nel suo significato matematico tradizionale, in quan-

to dati gli insiemi X, Y e Z, che non devono essere necessariamente distinti, si identifica con il simbolo  $R(X, Y, Z)$  una relazione su questi insiemi.

Se questa risulta essere a sua volta un insieme di n-uple ordinate  $(x, y, z)$ , tali che x appartenga a X, y a Y e z a Z, la relazione  $R(X, Y, Z)$  è un sottoinsieme del prodotto cartesiano degli insiemi X, Y, Z e può essere estesa a un numero qualsiasi di insiemi di dati. Graficamente questa relazione è espressa in figura 10.

Con questo modello una situazione reale viene rappresentata con un insieme di relazioni indicato con il nome di "schema relazionale", in cui è possibile definire le seguenti operazioni fondamentali:

- selezione di una relazione;
- proiezione di una relazione;
- giunzione di due relazioni.

Le tre precedenti operazioni hanno in input una o più relazioni che appartengono allo schema relazionale e forniscono in output una relazione "istantanea" che non fa parte dello schema.

Combinando in maniera opportuna queste tre operazioni è possibile ottenere qualsiasi informazione memorizzata nel data-base.

Si prende ora meglio in esame ciascuna delle precedenti operazioni:

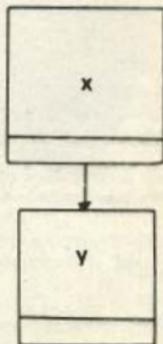


Fig. 5



- *selezione di una relazione.* In questo caso se si specifica una determinata condizione, si selezionano tutte le  $n$ -uple che la soddisfano. In una relazione  $R(X,Y)$  un'operazione di selezione può essere eseguita attribuendo un particolare valore a  $Y$ : selezione su  $R(X,Y)$  dove  $y_1=y_1'$  e si potrà ottenere anche un insieme vuoto di  $n$ -uple che soddisfa la condizione di partenza;

- *proiezione di una relazione.* Vengono eliminati degli attributi dalla relazione sopprimendo le eventuali  $n$ -uple coincidenti, come si può osservare dalla relazione in figura 11.

In essa si suppone di voler proiettare le relazioni su  $x,y$ , eliminando  $z$ , per ottenere la relazione rappresentata in figura 12;

- *giunzione di due relazioni.* Si suppone di avere due relazioni aventi uno o più elementi in comune, come ad esempio  $R(X,Y)$  e  $R(Y,Z)$ . L'operazione di giunzione consiste nel fondere le due relazioni senza che venga persa alcuna informazione.

## Un esempio

Nel caso della selezione dalla relazione

$R(\text{libri,collocazioni})$

l'operazione conduce ad una relazione

$R(\text{libri,collocazioni})$

che rappresenta un sottoinsieme dell'originale, in cui ogni  $n$ -upla sarà formata da

$(x,collocazione)$ .

Con la proiezione, invece, dalla relazione

$R(\text{autori,libri})$

con la sola chiave autori si genera una relazione

$R(\text{autori})$

di tipo semplice che fornisce solamente informazioni riguardanti gli autori.

Infine con l'operazione giunzione delle relazioni

$R1(\text{lettori,collocazioni})$

e  $R2(\text{libri,collocazioni})$

si genera una relazione

$R(\text{lettori,libri,collocazioni})$ .

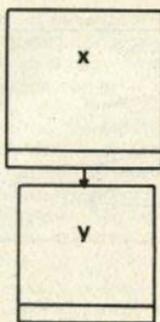


Fig. 6

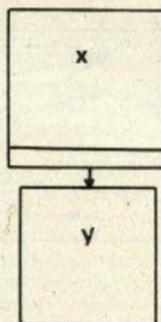


Fig. 6a

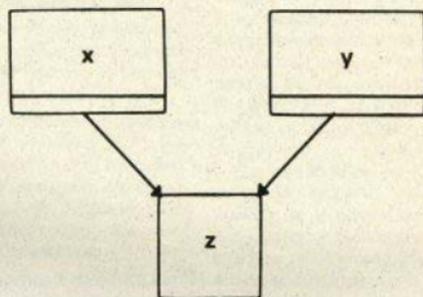


Fig. 7

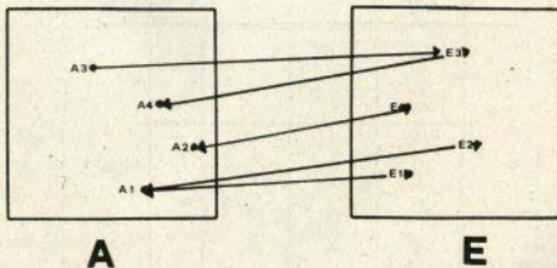


Fig. 8

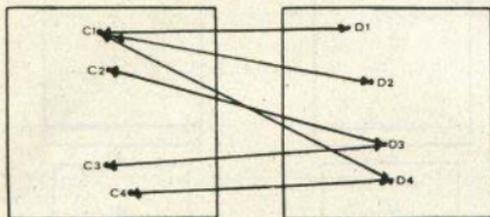


Fig. 9

X	Y	Z
X1	Y1	Z1
X2	Y2	Z2
X1	Y1	Z3
X3	Y2	Z1
X3	Y3	Z2
X3	Y3	Z1
X1	Y1	Z2

Fig. 10

X	Y
X1	Y1
X2	Y2
X1	Y1
X3	Y2
X3	Y3

Fig. 11

## L'evoluzione

A questo punto, nell'ambito di questo articolo, in cui si è cercato di spiegare che cosa sia e come è strutturato un data-base, si vuole dare un cenno su ciò che l'evoluzione rapidissima della componente comunicazione ha comportato sia per chi gestisce che per chi utilizza l'archivio. In origine il data-base costituiva lo strumento messo a disposizione degli operatori del centro di calcolo e veniva utilizzato dagli utenti per consultare i risultati ottenuti dalle applicazioni preparate dagli esperti.

Quando si giunse all'introduzione e alla diffusione dei terminali, sorsero anche la possibilità di fornire a utenti più o meno remoti gli stessi servizi che si potevano avere nel centro, e quindi si pervenne alla distribuzione delle funzioni in aree diverse sia geograficamente che logicamente, come esempio a uffici situati in luoghi diversi o a gruppi di utenti con differenti interessi e per applicazioni distinte.

Tutto questo portò quindi a una maggiore autonomia dell'utente in grado di effettuare scelte operative differenti. In quest'ottica il data-base rimane però centralizzato, in quanto le informazioni restano sempre nello stesso luogo, anche se risultano accessibili a distanze più o meno grandi.

La diminuzione dei costi dell'hardware ha introdotto la possibilità di cedere ad un certo numero di mini elaboratori la rete di comunicazione al fine di facilitare la gestione dei terminali. Come conseguenza di tutto ciò si ebbe la progettazione del Data Base Distribuito, il cui scopo era quello di gestire una parte delle informazioni arrivando ad avere per ogni mini-elaboratore, detto anche personal, un ambiente di lavoro con un proprio data-base per svolgere la maggior parte delle funzioni applicative.

## La progettazione

L'attività che porta alla progettazione di un data-base può essere strutturata in quattro fasi ben distinte fra loro:



- individuazione delle problematiche;
- progettazione dello schema concettuale;
- progettazione dello schema logico;
- progettazione dello schema fisico.

Le metodologie proposte nella risoluzione di questo problema si differenziano sia per il modo con cui vengono individuati i requisiti, tramite l'analisi dei dati o dei processi sui dati, che per l'approccio scelto per individuare i dati da esaminare.

Si deve osservare che attualmente i data-base sono organizzati fisicamente utilizzando un insieme di files con strutture che permettono l'accesso random alle informazioni. L'utilizzo del modello relazionale ha portato a sviluppare da parte degli studiosi forme diverse di memorizzazione che permettessero da una parte accessi più veloci alle informazioni e dall'altra minori occupazioni relative di memoria. Questi tipi di strutture fisiche sono:

- files invertiti;
- B-trees e B-trees.

Il file invertito è costituito da una chiave logico-fisica che permette la ricerca molto veloce delle informazioni all'interno della struttura del data-base. Infatti, considerando una relazione così come era stata definita quando si è parlato del

modello relazione, un record in un file invertito non corrisponde alla concezione convenzionale di record, bensì rappresenta l'insieme dei valori di un attributo.

I B-trees e B-trees permettono invece di memorizzare le informazioni utilizzando la struttura ad albero binario e multiplo che consenta la ricerca molto rapida e non hanno difficoltà, contrariamente alle liste invertite, di gestione durante l'aggiornamento.

E' molto importante osservare che la ricerca di nuove strutture fisiche ottimali sta perdendo terreno, in quanto l'orientamento attuale è rivolto alla realizzazione di Data Base Machines. Quest'ultima rappresenta la soluzione in termini hardware al problema dei DBMS.

Infatti, in questo caso, un elaboratore "back-end" funge da DBMS e viene interfacciato con l'elaboratore centrale in cui risiedono programmi applicativi.

In una configurazione di questo tipo ogni richiesta, anche se di tipo complesso, viene inserita nel DBM che la elabora accedendo in parallelo alle informazioni sul data-base, restituendo all'elaboratore centrale solo i dati che presentano le condizioni richieste.

In questo campo vi sono però ancora dei problemi non risolti di carattere essenzialmente elettronico riguardanti l'architettura parallela della Data Base Machine.

Infatti è proprio sul parallelismo elaborativo di questa, formato da multiprocessori, che si basa la buona efficienza della macchina stessa.

Vorrei, in chiusura di questo articolo in cui si è cercato di descrivere il data-base in maniera non certo completa data la complessità dell'argomento, esporre i problemi legati all'introduzione tecnologica di questa struttura aziendale, per applicazioni sia di tipo gestionale che scientifico su grossi e piccoli sistemi come archivio per antonomasia.

Inizialmente il data-base fu visto solo come una naturale evoluzione degli strumenti tecnici in possesso dell'EDP, e questo portò a una chiusura, per molto tempo, in una sfera esclusivamente dedicata ai soli addetti ai lavori, portando all'esclusione dell'utenza in genere. Solo nel momento in cui l'ambiente EDP ha fatto propri i concetti evoluti del data-base e sono stati risolti i problemi della sua governabilità, questo strumento è entrato con tutti i diritti nell'organizzazione aziendale.

E' questa la ragione per cui oggi sono sempre più frequenti i casi nei quali si adottano le tecnologie del data-base derivanti da considerazioni non solo tecniche ma anche organizzative, con il risultato di sviluppare analisi relative al settore interessato di conseguenza alla creazione di un modello per la rappresentazione della realtà l'ultimo del data-base.

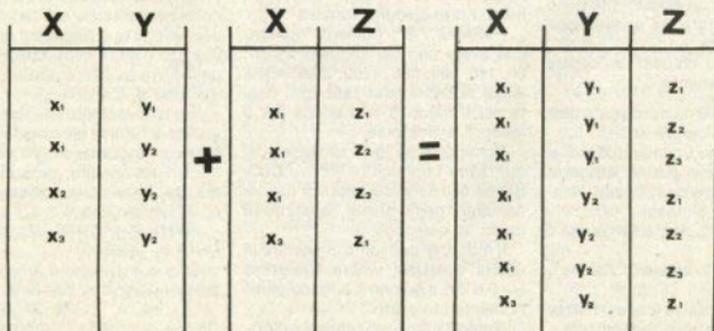


Fig. 12

Cominciamo la descrizione dei canali del sistema Spectrum + Interfaccia 1. Come saprete, l'Interfaccia 1 aggiunge (e gestisce) questi canali: i microdrives, l'interfaccia seriale, la network. Cominciamo con l'analizzare i microdrives.

## I microdrives

I comandi aggiunti e specifici per i microdrives sono:

**FORMAT "M"; numero; "nome";** formatta la cartuccia nel microdrive indicato da *numero* e la intesta con *nome*.

**ERASE; "M"; numero; "nome";** cancella il file chiamato *nome* contenuto nel microdrive *numero*.

**CAT numero;** mostra il catalogo dei programmi contenuti nel microdrive *numero*.

Altri comandi per i microdrives sono quelli già esistenti per le cassette, ma con sintassi estesa. Essi sono:

**SAVE \* "M"; numero; "nome";** salva il programma e le eventuali variabili sul microdrive *numero* e chiamandoli *nome*

**LOAD \* "M"; numero; "nome";** carica il programma e le eventuali variabili dal microdrive *numero* con *nome*

**VERIFY \* "M"; numero; "nome";** stessa funzione delle cassette.

**MERGE \* "M"; numero; "nome";** stessa funzione delle cassette

**OPEN # n; "M"; numero; "nome";** apre un canale *n* sul microdrive *numero* che si chiama *nome*; se il file *nome* non esiste il file viene creato in scrittura, se esiste viene aperto in lettura.

**CLOSE # n;** chiude un canale aperto in precedenza

**MOVE #n TO m;** devia i dati del canale *n* al canale *m*

Inoltre vi sono i comandi **SAVE** e **LOAD** che come per le cassette possono caricare array di dati. Hanno la seguente sintassi:

**SAVE \* "M"; numero; "nome"; a\$()**

**LOAD \* "M"; numero; "nome"; a\$()**

Il significato è: salva (carica) l'array **a\$()** dal microdrive *numero* chiamandolo con *nome*.

# I Canali dello Spectrum / II

di Damiano Bolla

È possibile salvare qualsiasi tipo di array sia numerico che di caratteri.

È naturalmente possibile salvare e caricare codice macchina o dati o in generale zone di memoria con la sintassi:

```
LOAD * "m"; numero; "nome"  
CODE n  
SAVE * "m"; numero; "nome"  
CODE n,m
```

che significa: carica il codice macchina dal microdrive *numero* con nome *nome* e mettilo a partire dalla locazione *n*; salva il codice macchina a partire da *n* e di lunghezza *m* nel microdrive *numero* con il nome *nome*.

## La porta seriale

Per prima cosa, quando si utilizza la seriale bisogna stabilire la velocità con la quale si vuole operare: ricordatevi che di solito questa è la maggior fonte di errori. Per settare il *baud rate* bisogna scrivere

**FORMAT "B"; numero.** *Numero* può avere uno dei seguenti valori: 50, 110, 300, 600, 1200, 2400, 4800, 9600, 19200; la *baud rate* specificata per il modo B vale anche per il modo T, e viceversa.

Generalmente con la seriale si utilizzano i comandi **OPEN** e **CLOSE**, però vi sono due modi di funzionamento degli stessi, appunto il modo 'B' e il modo 'T'.

Il modo 'B' consente di spedire in uscita qualsiasi valore compreso tra 0 e 255 e questo è il modo generalmente più usato.

Il modo 'T' invece consente di trasformare i token dello Spectrum in

caratteri ASCII, che sono stampabili dalla printer.

Per esempio, se volete stampare 10 REM ciao da damiano

se digitate  
**OPEN # 4; "B"**  
e poi date **LIST 4**, non vedete niente. Il fatto è che mancano alla stampante i codici 13 e 10, rispettivamente di stampa e di avanzamento linea; provate a spedirli scrivendo

**PRINT # 4; CHR\$ 13; CHR\$ 10;**

A questo punto avrete visto il vostro listato su carta, con una piccola differenza: dov'è la REM? Eseguite un reset, ridigite la linea di programma e scrivete

**OPEN # 3; "T": LIST 3**

Tutto OK? Avrete capito la differenza tra il modo 'b' e il modo 't'; la cosa carina è che si possono aprire in contemporanea, su due canali diversi, cioè:

**OPEN # 3; "T": OPEN 4; "B"**

In questo modo potete listare i programmi sul canale 3 e mandare codici di controllo sul canale 4. Per esempio si può mandare il codice 12 (il salto pagina nelle Epson e compatibili) in questo modo:

**PRINT # 4; CHR\$ 12;**

È così possibile attivare il modo grafico e tutte le altre opzioni che la stampante possiede. Se avete più numeri da spedire, nessun problema; per esempio la grafica si attiva così (sempre Epson & C.):

**PRINT # 4; CHR\$ 27; CHR\$ 75; CHR\$ n; CHR\$ m;**

dove *n* e *m* sono due numeri opportuni. Si può anche fare:

**LET a\$ = CHR\$ 27 + "K" + CHR\$n + CHR\$m: PRINT # 4; a\$**  
(Il cod. ASCII di "K" è 75).



La linea seriale dello Spectrum può essere usata per collegarsi con un modem, ma è necessario un programma in linguaggio macchina, perché il basic è lento e perde caratteri. Se il comando che viene utilizzato in prevalenza è OPEN non bisogna scordare che si può salvare e caricare da seriale con la sintassi.

```
LOAD "*"B"...
SAVE "*"B"..."
```

## La network

Passiamo alla network, che può essere considerata una seriale un po' particolare.

Per prima cosa dovete stabilire che numero di stazione volete essere voi, e per fare questo digitate

```
FORMAT "N"; numero
```

dove *numero* è il numero di stazione prescelto. Avete ora a disposizione il comando OPEN ulteriormente modificato

```
OPEN#n; "n"; numero
```

Questo significa che volete aprire un canale *n* verso la stazione *numero*. Ovviamente avrete poi a disposizione tutte le possibilità che un canale vi può dare. Potete salvare, caricare, verificare:

```
SAVE "*"N"; num
```

```
LOAD "*"N"; num
```

```
VERIFY "*"N"; num
```

Dove *num* è la stazione con la quale volete comunicare. Se *num* vale 0, le comunicazioni saranno di tipo generale, cioè trasmettete o ricevete comunicazioni a/da tutti i calcolatori collegati.

## Continuando...

L'interfaccia 1 possiede doti "nascoste", che si lasciano scoprire senza troppe difficoltà. Il segreto sta tutto nel saper utilizzare bene i comandi OPEN e specialmente MOVE.

Il comando OPEN si occupa di collegare un canale fisico con un canale logico e lo fa in maniera molto potente. Se per esempio avete costruito un programma che chiede tutti i suoi INPUT dal canale 4, cioè

```
INPUT # 4; a$
```

per esempio potete digitare

```
OPEN # 4, "k"
```

e il programma leggerà i suoi dati da tastiera, oppure

```
OPEN # 4; "b"
```

e il programma cercherà i suoi dati dalla linea seriale, o ancora

```
OPEN # 4; "m"; 1; "fileidat1"
```

Qui i dati sono scritti in un file creato in precedenza; ma non è finita:

```
OPEN # 4; "n"; 10
```

Ora i dati vengono chiesti allo Spectrum che ha 10 come numero di stazione della network.

Come potete vedere c'è di che divertirsi; lo stesso discorso si può fare per i dati in uscita. Se, per esempio, avete un programma database scritto in basic e alcuni amici che hanno il sistema Spectrum, potete divertirvi a collegare gli Spectrum tra di loro in modo che lavorino come dei terminali; la cosa funziona anche tra il QL e lo Spectrum.

Il comando MOVE serve per dirigere un flusso dati da una parte all'altra. Potete spostare un file di dati dal microdrive alla seriale oppure allo schermo o viceversa; eccovi alcuni esempi. Create un file che contenga un programma qualsiasi:

```
OPEN # 4; "m"; 1; "Fileidprova"
```

```
LIST # 4; CLOSE # 4
```

a questo punto

```
OPEN # 4; "m"; 1; "Fileidprova";
```

```
OPEN # 5; "t"
```

```
and infine
```

```
MOVE # 4 TO # 5
```

Vedrete come il file precedentemente creato vada a finire sulla stampante. Provate ancora, dopo aver chiuso e riaperto il canale 4

```
MOVE # 4 TO # 2
```

il file ora scorre sullo schermo.

Provate

```
MOVE # 0 TO # 2
```

Scrivete qualcosa, non disperatevi se non riuscite a uscire con BREAK (aspettate quando il computer vi chiederà di scrollare).

Se digitate OPEN # 2; "T" e avete la stampante collegata, qualsiasi cosa venga scritta sul video sarà spedita alla stampante.

Per leggere da un file vi sono diversi modi: il più utile è forse questo:

```
a$ = INKEYS # n
```

dove *n* è il file in questione; in questo modo leggete praticamente

qualsiasi cosa vi sia nel file. Un altro modo è quello di utilizzare

```
INPUT # n; a$
```

oppure

```
INPUT # n; a
```

Il primo caso se volete leggere una stringa, il secondo per leggere un numero.

Da notare che questa sintassi è indipendente da che cosa sia collegata al canale specificato, microdrive, seriale o network. A questo punto entriamo un po' più nel dettaglio dell'interfaccia 1. Poiché delle variabili di sistema abbiamo già parlato (vedi SC n. 12 pag. 45), vediamo ora come vengono gestite le varie periferiche in memoria.

Se avete seguito le puntate sull'accesso random ai microdrives avete visto che si faceva riferimento a una tabella che veniva creata quando si apriva un file su microdrive: eccovela qui di seguito per intero.

Lunghezza massima 595 bytes, indirizzo di partenza nel registro IX del processore. Il numero indica il byte progressivo.

0 Indirizzo 8

2 Indirizzo 8

4 "M"

5 Indirizzo delle subroutines di output nella ROM.

7 Indirizzo delle subroutines di input della ROM.

9 Indirizzo 595

11 CHBYTE Contatore del byte corrente.

13 CHREC Contatore del record corrente.

14 CHNAME Nome del file.

24 CHFLAG Flag di stato :255 = scrittura, 254 = lettura

25 CHDRIV Numero del microdrive 0-7

26 CHMAP Indirizzo della mappa del microdrive corrente

28 CHMAP Primi 12 bytes dell'installazione.

40 HDFLAG Byte di flag

41 HDNUMB Numero di settore utilizzato

44 HDNAME Nome della cartuccia.

54 HDCHK Byte di checksum.

55 Primi 12 bytes del blocco dati.

67 RECFLG Byte di flag 2 = EOF  
0 = normale





La proposta di realizzare un programma per il controllo di finestre video sullo Spectrum ha avuto un imprevisto successo, e abbiamo ricevuto numerose realizzazioni, sia in basic che in linguaggio macchina.

Abbiamo perciò deciso di pubblicarne ancora due, contrariamente alle consuetudini, dopo il software di S. Zendrini apparso sullo scorso numero; una citazione di merito anche per V. Mazzotta ed E. Sorcinelli, autori di implementazioni più che valide, che non pubblichiamo solo perché cinque versioni dello stesso programma sono decisamente troppe.

E arriviamo finalmente ai lavori migliori, inviati da B. Zamengo e C. Pancino, totalmente differenti tra di loro, ma entrambi altamente significativi nel loro particolare campo.

Il programma di B. Zamengo è infatti costituito da una efficientissima routine in linguaggio macchina, che viene usata molto semplicemente POKANDO i valori significativi (richiesti via INPUT) e chiamando la routine con una RAND USR: l'interfaccia utente è quindi ridotta al minimo indispensabile, ed è lasciata alla fantasia e alle necessità dell'utente, ma la routine è spettacolarmente veloce.

C. Pacino ha invece inviato un programma interamente scritto in basic: molto più lento del precedente per quanto riguarda il trasferimento delle finestre vero e proprio, ma con un'interfaccia utente (cioè il modo più o meno facile con cui l'utente finale si trova a operare) veramente eccezionale quanto a facilità e immediatezza d'uso.

Il consiglio che diamo è naturalmente quello di integrare i due programmi, sostituendo le linee del programma di Pancino che realizzano il trasferimento con la chiamata alla routine in ling. macchina dell'altro, per guadagnare in velocità mantenendo i vantaggi di un programma sofisticato come quello in basic di Claudio Pancino.

Per completezza, il pubblichiamo integralmente, segnalandovi solo che nel programma di Pancino le li-

## Window / II

nee che si occupano del trasferimento sono quelle da 3500 a 3820, per quanto riguarda il trasferimento da video a RAM, e da 3920 a 4660 per il trasferimento da RAM a video (in realtà, per quest'ultimo trasferimento le linee che operano materialmente il trasferimento sono quelle da 4500 in poi, ma prima c'è tutta la preparazione dei parametri necessari all'operazione).

### Il primo

di B. Zamengo

Il listato 1 è il programma basic utilizzato per caricare il codice in memoria ed eseguire un test della routine, tramite la stampa di una scritta ("prova") incorniciata, che costituisce la finestra da trasferire: premendo un tasto qualunque vengono richiesti i nuovi parametri (linea e colonna) della posizione dello schermo a cui va trasferita la finestra (che è stata salvata precedentemente in RAM). Terminato il test si possono cancellare le linee da 25 a 43, che disegnano la scritta di prova, mentre i parametri usati dalla routine per funzionare sono contenuti nelle locazioni:

57344-57345: (linea 50) contengono il valore 57790 che rappresenta l'indirizzo a partire dal quale va memorizzata la finestra prescelta.

57346: (linea 60) contiene il numero di riga del carattere in alto a sinistra della finestra.

57347: (linea 70) contiene il numero di colonna del carattere nell'angolo in alto a sinistra della finestra.

57348: (linea 80) contiene il numero di righe corrispondente all'altezza della finestra.

57349: (linea 90) contiene il numero di colonne corrispondente alla larghezza della finestra.

Il trasferimento (o salvataggio) della finestra dallo schermo alla memoria (a partire dall'indirizzo 57790) avviene con la chiamata RAND USR 57359, naturalmente dopo aver inizializzato le locazioni di memoria sopra riportate con i valori desiderati. Dopo avere stabilito in quale posizione dello schermo va spostata la finestra salvata, con delle POKE ai soliti valori 57346 e 57347, il trasferimento dalla RAM allo schermo si ottiene con la chiamata RAND USR 57707.

### Il secondo

di C. Pancino

Questo programma basic consente di scegliere tra 2 opzioni di lavoro: la prima permette di operare su una sola immagine dalla quale ricavare una sola finestra, la seconda consente di ricavare due finestre da una oppure due immagini originali.

In entrambe le opzioni, per selezionare una finestra è sufficiente spostare delle linee orizzontali e

verticali, che rappresentano i contorni della finestra (alto, basso, sinistro, destro); le linee vengono spostate mediante i soliti tasti con le frecce, mentre se spostandosi sul disegno con le linee si venissero a modificare i colori dell'inchiostro del disegno potremo ripristinarli con il comando "i" (ink); ogni fase di definizione delle righe è chiusa con il comando "f" (fine), che riporta l'utente nello stato di attesa comandi, segnalato con un BEEP.

La fase di definizione della finestra è chiusa col comando "c" (copia), che opera l'effettiva copia della finestra in RAM.

Il programma e le sue opzioni sono molto semplici da usare, essendo anche fornite di commenti e spiegazioni "in linea": così le procedure di rilocazione sullo schermo dell'immagine sono molto semplici, visualizzando lo spazio occupabile dalla finestra sullo schermo e consentendone lo spostamento con i tasti delle frecce; il comando "s" stamperà la finestra sul video, alla posizione indicata, mentre nel caso si fossero salvate 2 finestre in memoria si potrà procedere alla rilocazione della seconda, sempre con la prima fissa sullo schermo.

Altri comandi consentono il salvataggio su nastro dell'immagine video elaborata, la rilocazione delle finestre (se non si è ottenuto un effetto gradito) e la modifica degli attributi.



## Astrolabio

di Fabio Gallerini

Un astrolabio è un particolare strumento, attraverso il quale è possibile avere una rappresentazione del cielo e delle sue costellazioni in un particolare periodo; normalmente è costituito da due dischetti di carta sovrapposti che danno un'idea abbastanza approssimativa della volta celeste.

Ora invece, col programma che pubblichiamo, potremo avere l'immagine precisa del cielo, rigorosamente valida in ogni orario per la zona da cui osserviamo, dando modo agli appassionati di avere riferimenti esatti e ai profani di acquisire familiarità con le stelle, sia con l'osservazione diretta guidata dal computer (verificando cioè all'esterno) sia con l'apprendimento a tavolino, comodo anche perché può essere effettuato con qualsiasi condizione meteorologica...

L'uso del programma è semplicissimo; prima di farlo girare occorre però fornire allo Spectrum la posizione geografica da cui osserviamo, assegnando alle variabili LT (latitudine) e LN (longitudine) i valori opportuni, secondo le coordinate del nostro paesello, deducibili da qualsiasi carta geografica, sebbene con un certo margine di errore: con una carta topografica ricaverete i dati con la massima precisione.

La variabile FUSO (stessa riga) si riferisce al fuso orario e vale uno per l'Italia: se contate di osservare il cielo dalle montagne della Cocinina o del Perù modificate opportunamente.

Funzionamento del programma: vengono richiesti in input la data e l'ora dell'osservazione, quindi viene disegnata la mappa del cielo, costellazione per costellazione (ogni volta viene indicata in basso quale si sta disegnando); durante questa fase è possibile, premendo un tasto qualsiasi, interrompere l'esecuzione del programma per tutto il tempo in cui sarà tenuto premuto il tasto, permettendoci così di contemplare con più calma la volta celeste.

Le stelle sono di splendore variabile a seconda della loro lucentezza reale, e sono comprese in un cerchio, che rappresenta l'orizzonte terrestre, con il NORD in alto, l'EST a destra e così via; lo ZENIT (cioè il punto del cielo esattamente sopra la nostra testa) è al centro del cerchio.

A questo punto possiamo fare tre cose:

- 1) COPY del video su stampante;
- 2) cambiamento di ora e/o data per una nuova mappa.
- 3) ingrandimento di una zona della mappa attuale.

Questa è una delle caratteristiche più interessanti del programma: per ottenere l'ingrandimento bisogna specificare il fattore di ingrandimento (da 1 in su) e il punto che dovrà essere al centro della zona da ingrandire. Per fare ciò dovremo inserire le coordinate, e potremo farlo in uno qualsiasi dei sistemi di coordinate riconosciute dal programma:

le coordinate *altazimutali*, che in-



dividuano un punto del cielo semplicemente tramite la sua ALTEZZA in gradi sull'orizzonte, (da 0 a 90) e al suo AZIMUT, cioè l'angolo tra la direzione NORD, che ha azimut zero, e il punto in esame, girando verso est (l'azimut è l'angolo leggibile sulla bussola);

o le coordinate *equatoriali*, più difficili da trattare e quindi più dedicate ai veri appassionati, per le quali vi rimandiamo a un manuale di astronomia.

Per quanto riguarda il listato, c'è poco da dire perché è molto ben strutturato e autoesplicativo. È possibile inserire altre costellazioni oltre quelle disponibili (circa 30) sostituendo il 1000 dell'ultimo DATA riferito alle costellazioni (linea 1350) con 225, inserendo il nome della costellazione, le 5 coordinate *equatoriali* per ogni stella, e chiudendo l'ultimo DATA inserito con un 1000.

## Simon

di Alessandro Pardi

Credete di avere buona memoria visiva e auditiva? Anche dei buoni riflessi? Sicuramente allora o siete una rarità o non avete mai giocato a Simon.

Per chi, beato lui, non avesse mai sentito nominare questo perfido gioco, praticabile solitariamente o in gruppo, ricordiamo le elementarissime regole: ci sono quattro settori colorati diversamente, che il computer "accende" (facendoli brillare) secondo sequenze causali; a ogni settore è associato un beep di diversa tonalità, ed è parimenti associato un tasto.

Premendo nel giusto ordine i tasti dovete ripetere esattamente la sequenza che vi è stata proposta, e che si allunga ogni volta di una nota (e di un lampeggio colorato).

Più sequenze riuscite a ripetere esattamente, più aumenta la velocità di esecuzione; i tre livelli di difficoltà riguardano anch'essi variazioni di velocità e durata del beep.

La vittoria, assai improbabile, si trova oltre la cinquantesima *manche*.

Il programma contiene una breve routine in linguaggio macchina, per un'efficace accensione e spegnimento dei settori colorati.

## Slot machine

di Paolo De Rosa

Il programma è realizzato in due parti, per poter girare anche sul 16k.

La prima parte riguarda la grafica: dopo averla battuta, salvatela sul nastro, date RUN (vengono definiti gli UDG) e quindi NEW.

Ora digitate la seconda parte, senza alterare i numeri di linea (su di essi è basato il sistema di probabilità di uscita di un simbolo piuttosto che di un altro).

Il risultato è apprezzabile, considerato che il programma è tutto in basic.

Se non vi interessano nemmeno le vere slot-machine, lasciate perdere questo programma; se invece siete un incallito giocatore, potreste raggiungere lo stato di inebetimento estatico-annoiato caratteristico degli slot-dipendenti. Peccato solo allo Spectrum manchi la fessura per la monetina.

KHF TIRRADIO

Il più grande Network Italiano

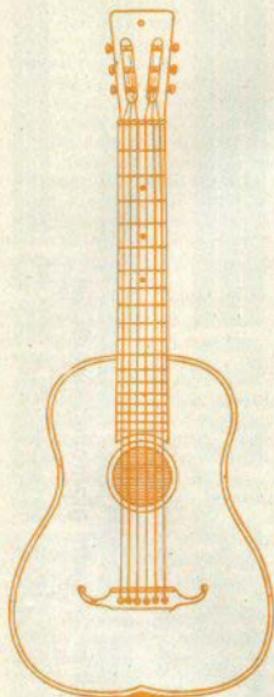
100 RADIO

6 ore di programmi giornalieri simultanei

CIRCUITO RADIO 84

CONCESSIONARIA PER LA PUBBLICITÀ

Via Vespri Siciliani 38 - 20146 Milano - Tel. 02/4226065-4224469-4231923-4225876



# Accordi per la chitarra

di Mario Donati

Un programma didattico costruito su un'idea decisamente buona, e abbastanza ben realizzato.

L'idea è quella di usare lo Spectrum come "istruttore ausiliario" per l'apprendimento della chitarra: il maestro in carne e ossa non è sempre a disposizione, tantomeno gratis... ecco perciò il fido computer che ci offre un divertente modo di ripassare gli accordi.

Il menu principale ha 5 opzioni:

- 1 - istruzioni per l'uso del programma;
- 2 - giri armonici: scegliete il "giro" che interessa tra quelli elencati, e vi viene mostrata la sequenza di accordi che lo compongono,

disegnando sullo schermo la sezione di tastiera con l'esatta posizione delle dita;

3 - accordatura: una serie di istruzioni visive e sonore per dare una registrazione alla vostra Fender;

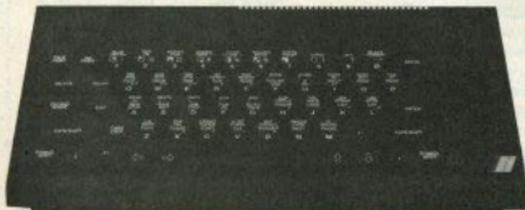
4 - preparazione: qualche chiarimento su come interpretare la tastiera disegnata sul video senza slogarsi le dita;

5 - accordi: il programma memorizza circa 90 accordi, sufficienti per servirci le orecchie dei vostri conviventi nella maggior parte delle situazioni; quando li saprete perfettamente a memoria, potrete sempre espandere il programma con altri accordi o altre opzioni.

## **CENTRO 2**

**ASSISTENZA HARDWARE**

**V. FRA CRISTOFORO, 2 - 20142 MILANO - (02) 8434368**



**CENTRO ASSISTENZA  
AUTORIZZATO**

**sinclair**  
MICROCOMPUTER



# Window/II

50

```
160 POKE 57347,V
170 CLS
180 RANDOMIZE USR 57707: REM
ROUTINE RAM -> VIDEO
```

5 REM

```
*****
*
* di BENIAMINO ZAMENGO *
* Via G. Mazzini 2 *
* 30030 Salzano (VE) *
*
*****
```

10 REM

TRASFERIMENTO E RITORNO

```
15 CLEAR 57343
20 GO SUB 9000
25 PRINT AT 9,5; "-----"
30 PRINT AT 10,5; "*" *
";AT 11,5;"* " *";AT 12,5
;"* " *
" 35 PRINT AT 12,5;"-----"
"
38 PLOT 51,96: : DRAW 8,0: DRA
W 0,-8: DRAW -8,0: PLOT 51,96: D
RAW 0,-16
40 PLOT 67,80: DRAW 0,16: DRAW
8,0: DRAW 0,-8: DRAW -8,0: DRAW
8,-8
41 PLOT 83,80: DRAW 0,16: DRAW
8,0: DRAW 0,-16: DRAW -8,0
42 PLOT 99,96: DRAW 4,-16: DRA
W 4,16
43 PLOT 115,80: DRAW 0,16: DRA
W 8,0: DRAW 0,-16: PLOT 116,80:
DRAW 7,0
50 POKE 57345,INT (57790/256):
POKE 57344,(57790-256*PEEK 5734
5): REM DESTINAZIONE
60 POKE 57346,9: REM RIGA
70 POKE 57347,5: REM COLONNA
80 POKE 57348,4: REM NUM.RIGHE
90 POKE 57349,12: REM NUM.COLO
NNE
100 RANDOMIZE USR 57359: REM
ROUTINE VIDEO -> RAM

110 PRINT #1:"PREMI UN TASTO PE
R CONTINUARE": PAUSE 0
120 POKE 57345,INT (57790/256):
POKE 57344,(57790-256*PEEK 5734
5): REM INDIRIZZO INIZIALE DEL B
LOCCO
130 INPUT "NUOVA RIGA = ";V: IF
v>(22-PEEK 57348) THEN BEEP .2
3,23: PRINT #1;AT 0,0;"troppo in
basso.premi un tasto": PAUSE 0:
INPUT "": GO TO 130
140 POKE 57346,V
150 INPUT INVERSE 1:"NUOVA COL
ONNA = ";V: IF v>(32-PEEK 57349)
THEN BEEP .2,22: PRINT #1;AT
1,0;"troppo a destra.premi un ta
sto": PAUSE 0: INPUT "": GO TO 1
```

190 GO TO 130

```
9000 LET f=57359: LET t=57789: F
OR n=f TO t: READ x: POKE n,x: N
EXT n: RETURN
```

```
9001 DATA 195,233,224,058,002,22
4,254,008
9002 DATA 048,004,014,000,024,01
4,254,016
9003 DATA 048,006,214,008,014,00
1,024,004
9004 DATA 014,002,214,016,050,00
6,224,071
9005 DATA 167,062,000,040,004,19
8,032,016
9006 DATA 252,033,003,224,134,09
5,022,000
9007 DATA 033,000,064,025,121,16
7,040,008
9008 DATA 017,000,008,061,040,00
1,025,025
9009 DATA 034,007,224,033,000,00
0,058,002
9010 DATA 224,167,040,007,071,01
7,032,000
9011 DATA 025,016,253,017,000,08
8,025,237
9012 DATA 075,003,224,006,000,00
9,034,009
9013 DATA 224,022,001,237,075,00
4,224,033
9014 DATA 006,224,070,033,011,22
4,062,008
9015 DATA 144,185,048,011,119,02
0,035,006
9016 DATA 000,237,068,129,079,02
4,239,113
9017 DATA 122,050,014,224,201,03
3,002,224
9018 DATA 237,091,000,224,001,01
3,000,237
9019 DATA 176,237,083,000,224,20
1,197,070
9020 DATA 229,197,221,229,253,22
5,006,008
9021 DATA 197,253,229,225,237,07
5,005,224
9022 DATA 006,000,237,176,001,00
0,001,253
9023 DATA 009,193,016,236,014,03
2,221,009
9024 DATA 193,016,222,001,000,00
7,221,009
9025 DATA 225,035,193,016,209,20
1,197,237
9026 DATA 075,005,224,006,000,23
7,176,014
9027 DATA 032,221,009,221,229,22
5,193,016
```



```

9028 DATA 237,201,205,018,224,20
5,148,224
9029 DATA 058,014,224,071,014,00
0,237,091
9030 DATA 000,224,033,011,224,22
1,042,007
9031 DATA 224,205,165,224,237,07
5,003,224
9032 DATA 014,000,042,009,224,22
1,042,009
9033 DATA 224,205,213,224,201,04
2,000,224
9034 DATA 035,035,017,004,224,00
1,011,000
9035 DATA 237,176,201,197,253,07
0,000,253
9036 DATA 229,197,221,229,253,22
5,006,008
9037 DATA 197,253,229,209,237,07
5,005,224
9038 DATA 006,000,237,176,001,00
0,001,253
9039 DATA 009,193,016,236,014,03
2,221,009
9040 DATA 193,016,222,001,000,00
7,221,009
9041 DATA 253,225,253,035,193,01
6,204,201
9042 DATA 197,237,075,005,224,00
6,000,237
9043 DATA 176,014,032,221,009,22
1,229,209
9044 DATA 193,016,237,201,042,00
0,224,070
9045 DATA 058,002,224,184,032,01
3,035,070
9046 DATA 058,003,224,184,032,00
5,205,020
9047 DATA 225,024,024,042,000,22
4,035,035
9048 DATA 126,050,004,224,035,12
6,050,005
9049 DATA 224,205,018,224,042,00
0,224,017
9050 DATA 013,000,025,058,014,22
4,071,014
9051 DATA 000,221,042,007,224,25
3,033,011
9052 DATA 224,205,034,225,237,07
5,003,224
9053 DATA 014,000,237,091,009,22
4,221,042
9054 DATA 009,224,205,087,225,20
1,000,000

```

```

3 DIM X(4): DIM Y(4): DIM R(4
): DIM S(4): DIM W(5)
9 REM Premesse con opzioni
10 PRINT AT 1,7; PAPER 2;"FINE
STRA SUL DISPLAY"; INVERSE 1;AT
4,0;"OPZIONI:"
15 PRINT AT 6,0; FLASH 1;"1";
FLASH 0;". Caricamento di un'im
agine video completa da cas
setta
Riarrangiamento di un
a fine-
stra da detta immagin
e e co-
pia in Ram.
Possibilita' di copia
finestra su cassetta"
20 PRINT FLASH 1;"2"; FLASH 0
;". Riarrangiamento di due diver
se finestre da una o due di
stinte immagini video e copi
e in RAM.
Possibilita' di rilocazione
per entrambe le finestre sul
lo schermo e copia dell'inna
gine globale su cassetta"
25 IF INKEY#="1" THEN LET K=1
: GO TO 40
30 IF INKEY#="2" THEN LET K=2
: GO TO 40
35 GO TO 25
40 CLS : LET flag=0
45 REM Diagramma a blocchi per
le istruzioni
50 PRINT AT 1,7; PAPER 2;"FINE
STRA SUL DISPLAY"
60 PLOT 48,175: DRAW 176,0: DR
AW 0,-24: DRAW -176,0: DRAW 0,23
70 PRINT AT 5,0;"PER MUOVERE L
E LINEE ORIZZONTALI VERTICALI S
EGUIRE LE INDICA- ZIONI DEL SEG
UENTE SCHEMA:"
80 PRINT AT 9,11; PAPER 3;"IST
RUZIONI"
90 PRINT AT 11,0;"LINEE
MOVIM. STOP ROUT"
100 PRINT AT 13,0;"S=sinistra
5-8"
110 PRINT "D=destra
F=fine"
120 PRINT "A=alto"
130 PRINT "B=basso 6-7""
C=copia"
135 PRINT "I=ink"
140 PLOT 88,72: DRAW 4,0: DRAW
0,-15: DRAW -4,0: PLOT 92,68: DR
AW 7,0: PLOT 88,55: DRAW 4,0: DR
AW 0,-15: DRAW -4,0: PLOT 92,44:
DRAW 7,0
150 PLOT 136,72: DRAW 4,0: DRAW
0,-32: DRAW -4,0: PLOT 140,60:
DRAW 54,0
160 PLOT 200,48: DRAW 0,-4: DRA
W 48,0: DRAW 0,4: PLOT 224,44: D
RAW 0,-22: DRAW -160,0
170 PRINT FLASH 1;AT 21,2;"PRE

```

1 REM  
 FINESTRA SUL DISPLAY  
 di PANCINO Claudio L.

(c) 1985

2 BORDER 1: PAPER 1: INK 7: C  
 LS : POKE 23609,50: POKE 23658,0  
 : LET D=58624: LET P=D: LET V#="



```

MI ~ENTER~ PER CONTINUARE"
180 IF CODE INKEY#=13 THEN GO
TO 200
190 GO TO 180
199 REM Caricamento dell'immagi
ne
200 CLS : PRINT "ISTRUZIONI PER
IL CARICAMENTO DELL'IMMAGINE
VIDEO:"
210 PRINT AT 4,0;"- POSIZIONA I
L NASTRO"'"-" CONTROLLA INSERIME
NTO JACK"'"-" PREMERE ~I~ PER IS
TRUZIONI"'"-" PREMERE ~R~ PER CA
RICAMENTO"'"-" AVVIA IL REGISTRA
TORE"
220 IF INKEY#="i" THEN CLS : 6
0 TO 50
230 IF INKEY#="r" THEN GO TO 2
50
240 GO TO 220
250 LOAD ""$SCREEN$
300 REM Definiz. var. linee
310 LET X(1)=0: LET X(2)=255: L
ET X(3)=0: LET X(4)=255
320 LET Y(1)=0: LET Y(2)=175: L
ET Y(3)=0: LET Y(4)=175
330 FOR I=1 TO 2
340 PLOT OVER 1;0,Y(I): DRAW
OVER 1;255,0: PLOT OVER 1;X(I),
0: DRAW OVER 1;0,175
350 NEXT I
360 POKE 23659,2: PRINT #0;"Alt
o Basso Sin Dex Fin Copia Ink"
399 REM Routin spostamento line
e
400 IF INKEY#="s" THEN GO TO 5
00
410 IF INKEY#="d" THEN GO TO 1
000
420 IF INKEY#="b" THEN GO TO 1
500
430 IF INKEY#="a" THEN GO TO 2
000
440 IF INKEY#="c" THEN GO TO 3
000
450 IF INKEY#="i" THEN INPUT "
COLORE INK : ";A: INK A: GO TO 3
60
455 BEEP .02,0
460 GO TO 400
499 REM linea verticale sinistr
a
500 BEEP 0.05,30: LET X(1)=X(3)
510 IF INKEY#="5" THEN LET X(3
)=X(3)-8: GO TO 550
520 IF INKEY#="8" THEN LET X(3
)=X(3)+8: GO TO 550
530 IF INKEY#="f" THEN BEEP .0
5,20: GO TO 400
540 GO TO 510
550 IF X(3)<0 THEN LET X(3)=0
560 IF X(3)>=X(2) THEN LET X(3
)=X(2)-8
600 PLOT OVER 1;X(1),0: DRAW

```

```

OVER 1;0,175: PLOT OVER 1;X(3),
0: DRAW OVER 1;0,175
610 GO TO 500
999 REM linea verticale destra
1000 BEEP 0.05,30: LET X(2)=X(4)
1010 IF INKEY#="5" THEN LET X(4
)=X(4)-8: GO TO 1050
1020 IF INKEY#="8" THEN LET X(4
)=X(4)+8: GO TO 1050
1030 IF INKEY#="f" THEN BEEP .0
5,20: GO TO 400
1040 GO TO 1010
1050 IF X(4)>=255 THEN LET X(4)
=255
1060 IF X(4)<=X(3) THEN LET X(4
)=X(3)+8
1100 PLOT OVER 1;X(2),0: DRAW
OVER 1;0,175: PLOT OVER 1;X(4),
0: DRAW OVER 1;0,175
1110 GO TO 1000
1499 REM linea orizzontale infer
iore
1500 BEEP 0.05,30: LET Y(1)=Y(3)
1510 IF INKEY#="6" THEN LET Y(3
)=Y(3)-8: GO TO 1550
1520 IF INKEY#="7" THEN LET Y(3
)=Y(3)+8: GO TO 1550
1530 IF INKEY#="f" THEN BEEP .0
5,20: GO TO 400
1540 GO TO 1510
1550 IF Y(3)<0 THEN LET Y(3)=0
1560 IF Y(3)>=Y(2) THEN LET Y(3
)=Y(2)-8
1600 PLOT OVER 1;0,Y(1): DRAW
OVER 1;255,0: PLOT OVER 1;0,Y(3
): DRAW OVER 1;255,0
1610 GO TO 1500
1999 REM linea orizzontale super
iore
2000 BEEP 0.05,30: LET Y(2)=Y(4)
2010 IF INKEY#="6" THEN LET Y(4
)=Y(4)-8: GO TO 2050
2020 IF INKEY#="7" THEN LET Y(4
)=Y(4)+8: GO TO 2050
2030 IF INKEY#="f" THEN BEEP .0
5,20: GO TO 400
2040 GO TO 2010
2050 IF Y(4)>=175 THEN LET Y(4)
=175
2060 IF Y(4)<=Y(3) THEN LET Y(4
)=Y(3)+8
2100 PLOT OVER 1;0,Y(2): DRAW
OVER 1;255,0: PLOT OVER 1;0,Y(4
): DRAW OVER 1;255,0
2110 GO TO 2000
3000 INK 7: FOR I=3 TO 4
3010 PLOT OVER 1;X(I),0: DRAW
OVER 1;0,175: PLOT OVER 1;0,Y(I
): DRAW OVER 1;255,0
3020 NEXT I
3050 PRINT #0; FLASH 1:" Copia
Ram in fase operativa "
3499 REM copia ram
3500 LET X3=INT (X(3)/8): LET X4

```

```

=INT (X(4)/8): LET Y3=-INT (Y(3)
/8)+21: LET Y4=-INT (Y(4)/8)+21:
LET D1=INT ((X(4)-X(3))/8): LET
D2=INT ((Y(4)-Y(3))/8)
3510 LET C=(1 AND Y3<8)+(2 AND Y
3>=8 AND Y3<16)+(3 AND Y3>=16)
3520 LET B=(1 AND Y4<8)+(2 AND Y
4>=8 AND Y4<16)+(3 AND Y4>=16)
3530 LET G=(Y4 AND Y4<8)+(Y4-8 A
ND Y4)>=8 AND Y4<16)+(Y4-16 AND Y
4)>=16)
3540 LET G1=(Y3 AND Y3<8)+(Y3-8
AND Y3)>=8 AND Y3<16)+(Y3-16 AND
Y3)>=16)
3550 LET E=16384+2048*(B-1)+(G*3
2)+X3: LET F=E: LET L=E
3560 FOR M=B TO C
3570 LET H1=(G+1 AND B=M)+(1 AND
B<M)
3580 LET H2=(G1+1 AND C=M)+(B AN
D C>M)
3590 FOR I=H1 TO H2
3600 FOR X=1 TO 8
3610 FOR N=E TO E+D1
3620 POKE D,PEEK N
3630 LET D=D+1
3640 NEXT N
3650 LET E=E+256
3660 NEXT X
3670 LET L=L+32: LET E=L
3680 NEXT I
3690 LET F=F+32*(H2-H1+1)+1792:
LET E=F: LET L=E
3700 NEXT M
3749 REM copia attributi
3750 LET J=22528+(Y4*32)+X3
3760 FOR M=1 TO D2+1
3770 FOR X=J TO J+D1
3780 POKE D,PEEK X
3790 LET D=D+1
3800 NEXT X
3810 LET J=J+32
3820 NEXT M
3900 CLS : PRINT AT 8,0:"*****
*****":AT 10
,6:"COPIA RAM COMPLETATA""****
*****"
3901 IF K=1 THEN GO TO 3910
3903 IF K=2 AND flag=0 THEN LET
T=D: LET T1=D1: LET T2=D2: PAUS
E 100: CLS : PRINT "ISTRUZIONI P
ER IL CARICAMENTO DELLA STESSA
O DI UNA SECONDA DIVERSA IMMA
GINE VIDEO": LET flag=1: GO TO 2
10
3904 IF K=2 AND flag=1 THEN LET
T3=D1: LET T4=D2
3910 PRINT AT 21,2: FLASH 1:"PRE
MI ~ENTER~ PER CONTINUARE"
3911 IF CODE INKEY#=13 THEN GO
TO 3920
3912 GO TO 3911
3919 REM rilocalazione
3920 CLS : PRINT "PROCEDURA DI R

```

```

ILOCAZIONE DELLA FINESTRA VIDEO
IN POSIZIONE DE- SIDERATA"
3930 PRINT AT 5,11: PAPER 3:"IST
RUZIONI""
3940 PRINT "- COMPARSA DELLO SPA
ZIO OCCUPA- BILE DALLA FINESTRA
NELL'ANGOLO SUPERIORE SINISTRO"
"- POSSIBILITA' DI RILCAZIONE
SULLO SCHERMO DELLO SPAZIO E
QUINDI DELLA FINESTRA CON I TA
- STI DA 5 A 8""
3950 PRINT "="> Sinistra = 5""=">
Destra = 8""="> Alto = 7"
"="> Basso = 6""="> Stampa
= s"
3960 PRINT #0: FLASH 1:" PREMI
~ENTER~ PER CONTINUARE"
3970 IF CODE INKEY#=13 THEN GO
TO 4000
3980 GO TO 3970
4000 CLS : IF K=2 THEN LET flag
=0: LET D1=T1: LET D2=T2: LET P=
58624
4001 IF K=2 AND flag=1 THEN LET
P=T: LET D1=T3: LET D2=T4
4010 LET R(3)=0: LET R(4)=8*D1+7
: LET S(3)=175-(8*D2+7): LET S(4
)=175
4020 FOR I=3 TO 4
4030 PLOT OVER 1;0,S(I): DRAW
OVER 1;255,0: PLOT OVER 1;R(I),
0: DRAW OVER 1;0,175
4040 NEXT I
4045 IF K=1 OR K=2 AND flag=0 TH
EN PRINT #0:"5=sin 6=basso 7=al
to 8=dx Stampa"
4049 REM spostamento
4050 LET R(1)=R(3): LET R(2)=R(4
): LET S(1)=S(3): LET S(2)=S(4)
4060 IF INKEY#="8" THEN LET R(3
)=R(3)+8: LET R(4)=R(4)+8: GO TO
4130
4070 IF INKEY#="5" THEN LET R(3
)=R(3)-8: LET R(4)=R(4)-8: GO TO
4130
4080 IF INKEY#="6" THEN LET S(3
)=S(3)-8: LET S(4)=S(4)-8: GO TO
4200
4090 IF INKEY#="7" THEN LET S(3
)=S(3)+8: LET S(4)=S(4)+8: GO TO
4200
4100 IF INKEY#="s" THEN GO TO 4
300
4110 BEEP .02,30
4120 GO TO 4060
4130 IF R(3)<0 THEN LET R(3)=0:
LET R(4)=8*D1+7
4140 IF R(4)>255 THEN LET R(4)=
255: LET R(3)=255-(8*D1+7)
4150 FOR I=1 TO 4
4160 PLOT OVER 1;R(I),0: DRAW
OVER 1;0,175
4170 NEXT I
4180 GO TO 4050

```



# ilistabi

```
4200 IF S(3)<0 THEN LET S(3)=0:
LET S(4)=8*D2+7
4210 IF S(4)>175 THEN LET S(4)=
175: LET S(3)=175-(8*D2+7)
4220 FOR I=1 TO 4
4230 PLOT OVER 1;0,S(I): DRAW
OVER 1;255,0
4240 NEXT I
4250 GO TO 4050
4300 FOR I=3 TO 4
4301 PLOT OVER 1;R(I),0: DRAW
OVER 1;0,175: PLOT OVER 1;0,S(I)
: DRAW OVER 1;255,0
4302 NEXT I
4310 LET R3=INT (R(3)/8): LET R4
=INT (R(4)/8): LET S3=-INT (S(3)
/8)+21: LET S4=-INT (S(4)/8)+21
4320 LET C1=(1 AND S3<8)+(2 AND
S3>8 AND S3<16)+(3 AND S3>16)
4330 LET B1=(1 AND S4<8)+(2 AND
S4>8 AND S4<16)+(3 AND S4>16)
4340 LET G2=(S4 AND S4<8)+(S4-8
AND S4>8 AND S4<16)+(S4-16 AND
S4>16)
4350 LET G3=(S3 AND S3<8)+(S3-8
AND S3>8 AND S3<16)+(S3-16 AND
S3>16)
4500 LET E1=16384+2048*(B1-1)+(G
2*32)+R3
4510 LET F1=E1: LET L1=E1
4520 FOR M=B1 TO C1
4530 LET H3=(G2+1 AND B1=M)+(1 A
ND B1<M)
4540 LET H4=(G3+1 AND C1=M)+(8 A
ND C1>M)
4550 FOR I=H3 TO H4
4560 FOR X=1 TO 8
4570 FOR N=E1 TO E1+D1
4580 POKE N,PEEK P
4590 LET P=P+1
4600 NEXT N
4610 LET E1=E1+256
4620 NEXT X
4630 LET L1=L1+32: LET E1=L1
4640 NEXT I
4650 LET F1=F1+32*(H4-H3+1)+1792
: LET E1=F1: LET L1=E1
4660 NEXT M
4699 REM attributi
4700 LET J1=22528+(S4*32)+R3
4710 FOR M=1 TO D2+1
4720 FOR X=J1 TO J1+D1
4730 POKE X,PEEK P
4740 LET P=P+1
4750 NEXT X
4760 LET J1=J1+32
4770 NEXT M
4780 IF flag=1 THEN GO TO 4800
4790 IF K=2 AND flag=0 THEN LET
flag=1: GO TO 4001
4800 IF V$="m" THEN GO TO 7500
4999 REM copia su cassetta o ril
ocazione
5000 INPUT "
```

```
": PRINT #0;"Save Mo
difica attr. Rilocalazione"
5001 LET V$=INKEY$
5005 IF V$<>"s" AND V$<>"r" AND
V$<>"m" THEN GO TO 5001
5006 INPUT "
"
5010 IF V$="s" THEN GO TO 5500
5020 IF V$="r" THEN CLS : LET P
=58524: GO TO 3920
5030 IF V$="m" THEN GO TO 7000
5500 INPUT "TITOLO IMMAGINE: ";T
$
5510 IF LEN T$>10 THEN GO TO 55
00
5520 SAVE T$SCREEN$
5530 CLS : PRINT INVERSE 1;"OPZ
IONI": PRINT AT 3,0: FLASH 1;"1
"; FLASH 0;" . RITORNO MENU ""';
FLASH 1; INVERSE 0;"2"; FLASH 0;
". STOP"
5540 IF INKEY$="1" THEN GO TO 1
5550 IF INKEY$="2" THEN STOP
5560 GO TO 5540
7000 INPUT "INK (0-7): ";W(2),"P
APER (0-7): ";W(3)
7050 INPUT "BRIGHT (0-1):";W(4),
"FLASH (0-1): ";W(5)
7100 BORDER W(3): LET W(1)=W(2)+
8*W(3)+(64 AND W(4)=1)+(128 AND
W(5)=1)
7500 FOR X=22528 TO 23296
7510 POKE X,W(1)
7520 NEXT X: GO TO 5000
9998 SAVE "screen$" LINE 1
9999 VERIFY "screen$"
```

## Astrolabio

```
1 REM *****
2 REM ** ASTROLABIO **
3 REM ** **
4 REM ** BY **
5 REM ** **
6 REM ** GALLERINI FABIO **
7 REM *****
8 REM
10 REM LETTURA DATI
20 GO SUB 1670
30 REM ROUTINE PRINCIPALE
40 GO SUB 120
50 INK 7: PAPER 1: BORDER 1: C
LS : CIRCLE 132,88,85: LET q=.95
: LET x0=0: LET y0=0: GO SUB 420
60 PRINT #0;AT 1,0;"Ingrandime
nto Cambio data Stampa"
70 LET s$=INKEY$: IF s$<>"i" A
ND s$<>"s" AND s$<>"c" THEN GO
TO 070
80 IF s$="s" THEN COPY : GO T
O 060
```



```

90 IF S$="c" THEN GO TO 040
100 GO SUB 340: GO TO 060
110 REM DISEGNO
120 IF PEEK 50000=255 THEN GO
TO 210
130 POKE 50000,255: PAPER 6: IN
K 0: BORDER 6: CLS : FOR I=0 TO
13: PRINT PAPER 7;AT 2+I,5;"
": NEXT I
140 PLOT 39,159: DRAW 88,0: DRA
W 0,-112: DRAW -88,0: DRAW 0,112
: PLOT 40,158: DRAW 86,0: DRAW 0
,-110: DRAW -86,0: DRAW 0,110
150 FOR I=2 TO 6 STEP 2: PLOT 4
0+I,159+I: DRAW 88,0: DRAW 0,-11
2: NEXT I
160 CIRCLE 200,100,40: CIRCLE 2
00,100,34: PLOT 156,144: DRAW 87
,0: DRAW 0,-87: DRAW -87,0: DRAW
0,87
170 FOR I=2 TO 8: PLOT 156+I,14
4+I: DRAW 86,0: PLOT 243+I,143+I
: DRAW 0,-86: NEXT I
180 FOR I=0 TO 2*PI STEP PI/6
190 PLOT 32*COS I+200,32*SIN I+
100: DRAW 7*COS I,7*SIN I
200 NEXT I: CIRCLE 200,100,2: R
ANDOMIZE USR 50001
210 REM RICHIESTA DATI
220 INK 0: BORDER 6: BRIGHT 0:
RANDOMIZE USR 50013: PAPER 7: IN
PUT PAPER 7;AT 0,10;"GIORNO
":G: IF G<0 OR G>31 THEN GO TO
210
230 PRINT INK 2;AT 8,9+(G<10);
G
240 INPUT PAPER 7;AT 0,11;"MES
E ":MS: IF MS<1 OR MS>12 THEN
GO TO 240
250 PRINT AT 4,6;M$(MS)
260 INPUT PAPER 7;AT 0,10;"ANN
O ":A: IF A<100 THEN LET A=A+
1900
270 PRINT AT 11,8;A
280 INPUT PAPER 7;AT 0,11;"ORE
":O: INPUT PAPER 7;AT 0,10;"
MINUTI ":M: IF O>24 OR M>60 TH
EN GO TO 280
290 LET TML=O+M/60: PLOT 200,10
0: DRAW 22*SIN (TML*PI/6),22*COS
(TML*PI/6): PLOT 200,100: DRAW
29*SIN (M*PI/30),29*COS (M*PI/30
)
300 PRINT AT 21,5;"Confermi i d
ati (s/n)"
310 LET S$=INKEY$: IF S$<>"s" A
ND S$<>"n" THEN GO TO 310
320 IF S$="n" THEN GO TO 120
330 RETURN
340 REM DATI PER L'INGRANDIMEN
TO
350 INPUT "Introduci il fattore
di ingrandimento ":q
360 PRINT #0;AT 1,0;"Vuoi espri

```

```

mere il centro della zona da in
grandire in coordinate Equatoria
li o Altazimutali "
370 LET S$=INKEY$: IF S$<>"a" A
ND S$<>"e" THEN GO TO 370
380 IF S$="e" THEN INPUT AT 1,
4;"Ascensione (0-24) ";asc: INPU
T AT 1,4;"Declinazione (-90/90)
";dec: GO SUB 1360: GO SUB 1470:
GO SUB 1590: GO TO 400
390 INPUT AT 1,7;"altezza (0-90
) ";al: INPUT AT 1,7;"azimut (0-
360) ";az
400 LET X0=Q*(90-AL)*SIN (AGG*A
Z): LET Y0=Q*(90-AL)*COS (AGG*A
Z)
410 CLS : GO SUB 420: RETURN
420 REM STAMPA STELLE
430 RESTORE 620
440 GO SUB 1360: GO SUB 1470
460 READ AS
470 IF AS=1000 THEN RETURN
480 IF AS=255 THEN READ A$: PR
INT #0;AT 1,9;"
":AT 1,(32-LEN A$)/2;A$: BEEP
0.08,30: READ AS
490 READ US,DE,DU,MAG
500 IF INKEY$<>" " THEN GO TO 5
00
510 LET ASC=AS+US/60: LET DEC=D
E+DU/60*SGN DE-90
520 IF DEC<LT-180 THEN GO TO 4
60
530 LET P=AGG*15*(TSL-ASC)
540 LET X=Q*DEC*SIN P+132-X0
550 LET Y=Q*(DEC*COS P-LT+90)+8
8-Y0
560 IF (X-132)*(X-132)+(Y-88)*(
Y-88)>=8100*Q*Q THEN GO TO 460
570 IF (X<1 OR X>254 OR Y<1 OR
Y>174) THEN GO TO 460
580 IF MAG>3 THEN PLOT X,Y: GO
TO 460
590 IF MAG<1.65 THEN PLOT X,Y:
PLOT X,1+Y: PLOT X,Y-1: PLOT X+
1,Y: PLOT X-1,Y: GO TO 460
600 PLOT X,Y: PLOT X+1,Y: GO TO
460
610 REM COORDINATE STELLE
620 DATA 255,"Orsa minore"
630 DATA 14,51,74,22,2.24,15,21
,72,1,3,1,17,48,86,37,4,1,49,89,
2,2,1,255
640 DATA "Orsa maggiore"
650 DATA 12,52,56,14,1,7,11,16,
31,49,3,2,11,16,33,22,2,6,13,46,
49,34,1,9,11,1,62,1,1,9,12,13,57
,19,3,4,10,59,56,39,2,4,13,22,55
,11,1,9,11,51,53,58,2,5,8,56,48,
14,2,9,10,19,41,45,3,10,14,43,10
,3,255
660 DATA "Cassiopeia"
670 DATA 0,46,57,33,3,0,6,58,52
,2,4,0,53,60,26,2,1,23,59,59,3,0

```



# LISTA

,38,56,16,2,5,1,51,63,25,3,3,255  
680 DATA "Cefeo"  
690 DATA 21,17,62,22,2,6,21,28,  
70,20,3,3,23,37,77,21,3,2,255  
700 DATA "Ercole"  
710 DATA 17,12,14,27,2,5,16,4,2  
1,35,2,8,17,13,24,54,3,2,255  
720 DATA "Dragone"  
730 DATA 17,29,52,20,2,5,17,55,  
51,30,2,4,17,9,65,47,3,9,19,13,6  
7,34,3,2,14,3,64,37,3,255  
740 DATA "Andromeda"  
750 DATA 2,1,42,5,2,1,7,35,21,2  
255  
760 DATA "Gemelli"  
770 DATA 6,53,16,27,1,9,7,31,32  
0,1,6,6,41,25,11,3,2,7,42,28,9,  
1,2,7,17,22,5,3,5,6,12,22,31,2,9  
6,20,22,32,2,9,255  
780 DATA "Aquila"  
790 DATA 19,53,6,17,3,9,19,48,8  
44,8,19,23,3,35,3,4,19,44,10,2  
9,2,8,255  
800 DATA "Auriga"  
810 DATA 5,13,45,57,2,1,5,56,44  
57,1,9,255  
820 DATA "Lira"  
830 DATA 18,48,33,18,3,2,18,57,  
32,37,3,2,18,35,38,44,1,255  
840 DATA "Leone"  
850 DATA 10,14,23,40,3,5,11,12,  
15,42,3,3,11,47,14,51,2,2,9,43,2  
4,0,3,5,9,50,26,15,4,1,10,6,12,1  
3,1,3,11,11,20,48,2,6,255  
860 DATA "Cigno"  
870 DATA 20,44,33,47,2,6,19,50,  
44,35,2,8,19,54,35,1,3,1,19,29,2  
7,51,2,4,20,40,45,6,1,3,20,20,40  
6,2,3,255  
880 DATA "Pegaso"  
890 DATA 23,1,27,49,2,6,22,41,2  
9,58,3,23,2,14,56,2,6,22,39,10,3  
4,3,6,21,42,9,39,2,5,0,11,14,54,  
2,9,255  
900 DATA "Perseo"  
910 DATA 3,21,49,41,1,9,3,5,40,  
46,2,2,3,51,31,44,2,9,3,21,49,41  
1,95,255  
920 DATA "Eridano"  
930 DATA 2,56,-40,30,2,8,1,36,-  
57,29,0,6,5,5,-5,9,2,9,3,41,-9,5  
6,3,7,4,34,-30,40,4,1,3,56,-13,3  
9,3,2,255  
940 DATA "Scorpione"  
950 DATA 16,3,-19,40,2,9,16,26,  
-26,19,1,15,57,-22,29,2,17,30,-3  
7,4,1,7,255  
960 DATA "Cancro"  
970 DATA 8,56,12,3,4,2,8,42,18,  
20,4,1,8,40,21,39,4,7,255  
980 DATA "Vergine"  
990 DATA 11,48,2,3,3,8,12,39,-1  
11,3,8,12,53,3,40,3,7,13,23,-10  
54,1,2,255

1000 DATA "Aquario"  
1010 DATA 20,45,-9,41,4,21,29,-5  
48,3,22,52,-16,5,3,255  
1020 DATA "Corvo"  
1030 DATA 12,6,-24,27,4,2,12,27,  
-16,14,2,8,12,32,-23,7,2,8,12,8,  
-22,20,2,8,255  
1040 DATA "Idra"  
1050 DATA 9,25,-8,26,2,1,255  
1060 DATA "Boote"  
1070 DATA 14,13,19,27,-0,2,15,0,  
40,35,3,6,14,30,38,32,3,15,23,37  
33,4,2,255  
1080 DATA "Orione"  
1090 DATA 5,45,-9,41,2,2,5,52,7,  
24,0,8,5,34,-1,14,1,7,5,38,-1,58  
2,5,22,6,18,1,7,5,33,-5,56,4,5,  
32,9,54,4,5,29,0,0,2,4,5,12,-8,1  
5,0,3,255  
1100 DATA "Sagittario"  
1110 DATA 19,20,-40,43,4,19,19,-  
44,33,4,18,59,-29,57,2,4,18,21,-  
34,25,1,9,18,25,-25,27,2,9,18,18  
-29,51,2,8,18,3,-30,26,3,2,19,1  
-21,49,2,1,255  
1120 DATA "Pesci"  
1130 DATA 1,59,2,31,4,3,255  
1140 DATA "Cane maggiore"  
1150 DATA 6,57,-28,54,1,7,7,22,-  
29,12,2,4,6,18,-30,2,3,6,20,-17,  
56,2,7,1,-15,33,4,6,43,-16,39,-1  
4,7,6,-26,19,2,255  
1160 DATA "Serpente"  
1170 DATA 15,42,6,35,2,7,255  
1180 DATA "Cane da caccia"  
1190 DATA 12,31,41,38,4,3,12,54,  
38,35,2,6,255  
1200 DATA "Balena"  
1210 DATA 1,49,-10,35,3,9,0,41,-  
18,16,2,2,2,41,3,2,3,5,3,0,3,54,  
3,3,2,17,-3,12,3,255  
1220 DATA "Ariete"  
1230 DATA 3,9,19,32,4,2,4,23,14,  
2,2,1,51,19,3,4,1,52,20,34,2,8,2  
55  
1240 DATA "Capricorno"  
1250 DATA 20,18,-14,56,3,2,20,15  
-12,42,3,7,255  
1260 DATA "Pesce australe"  
1270 DATA 22,55,-29,53,1,3,255  
1280 DATA "Corona boreale"  
1290 DATA 15,33,26,53,2,3,255  
1300 DATA "Cane minore"  
1310 DATA 7,24,8,23,3,7,37,5,21,  
5,255  
1320 DATA "Ofiuco"  
1330 DATA 17,41,4,35,2,9,17,33,1  
2,36,2,17,8,-15,40,3,2,16,12,-3,  
34,4,255  
1340 DATA "Bilancia"  
1350 DATA 15,33,-14,37,4,14,48,-  
15,50,2,9,15,14,-9,12,2,7,1000  
1360 REM CALCOLO GIORNI PASSATI  
DALLO 0 0 1980

```

1370 IF MS>2 THEN GO TO 1420
1380 LET D=MS-1
1390 IF INT (A/4)=A/4 THEN LET
D=D*62: GO TO 1410
1400 LET D=D*63
1410 LET D1=INT (D/2)+6: GO TO 1
450
1420 LET D=INT (30.6*(MS+1))
1430 IF INT (A/4)=A/4 THEN LET
D1=D-62+G: GO TO 1450
1440 LET D1=D-63+G
1450 LET D=D1+365*(A-1980)+INT (
(A-1981)/4)+1
1460 RETURN
1470 REM CONVERSIONE TML IN TSG
1480 LET TMG=TML-FUSO
1490 IF TMG>24 THEN LET TMG=TMG
-24
1500 IF TMG<0 THEN LET TMG=TMG+
24
1510 LET T0=D1*AA-BB
1520 LET TSG=CC*TMG+T0
1530 IF TSG>24 THEN LET TSG=TSG
-24
1540 IF TSG<0 THEN LET TSG=TSG+
24
1550 LET TSL=TSG+LN/15
1560 IF TSL>24 THEN LET TSL=TSL
-24
1570 IF TSL<0 THEN LET TSL=TSL+
24
1580 RETURN
1590 REM CONVERSIONE COORDINATE
EQUATORIALI IN ALTAZIMUTALI
1600 LET HQ=(TSL-ASC)*15
1610 IF HQ<0 THEN LET HQ=HQ+360
1620 LET SD=SIN (DEC*AGG)
1630 LET AL=GGA*ASN (SD*SL+COS (
AGG*DEC)*CL*COS (AGG*HQ))
1640 LET AZ=GGA*ACS ((SD-SL*
SIN (AGG*AL))/CL/COS (AGG*AL))
1650 IF SIN (AGG*HQ)>0 THEN LET
AZ=360-AZ
1660 RETURN
1670 REM DATI
1680 DIM M$(12,9): LET LT=43.794
642: LET LN=13.111111: LET FUSO=
1
1690 LET AA=.0657098: LET BB=17.
409421: LET CC=1.002738: LET DD=
.997270
1700 LET AGG=PI/180: LET GGA=180
/PI: LET SL=SIN (AGG*LT): LET CL=
COS (AGG*LT)
1710 RESTORE 1720: FOR i=1 TO 12
: READ M$(i): NEXT i
1720 DATA " GENNAIO", "FEBBRAIO",
" MARZO", " APRILE", " MAGGIO",
GIUGNO", " LUGLIO", " AGOSTO", "SET
TEMBRE", " OTTOBRE", "NOVEMBRE", "D
ICEMBRE"
1730 RESTORE 1740: FOR i=0 TO 23
: READ b: POKE 5000+i,b: NEXT i
1740 DATA 33,0,64,17,32,203,1,0,

```

```

27,237,176,201,17,0,64,33,32,203
,1,0,27,237,176,201
1750 RETURN

```

## Simon

```

1 REM -----
2 REM -PROGRAM BY ALEX PARDI-
4 REM -----
5 LET TOP=0
6 PAPER 0: BORDER 0: INK 7: C
LS
7 PRINT AT 10,5: "BENVENUTO DA
SIMON!"
8 PRINT AT 12,5: "INPUT DIFFIC
OLTA' (1-3)"
9 PAUSE 0: LET K=CODE INKEY$:
IF K<49 OR K>51 THEN GO TO 8
10 LET LEV=K-48
20 LET ERR=0: LET RT=0: CLS :
GO SUB 9000
25 REM SEQUENZA COMPUTER
30 PAUSE 100: DIM A(50)
40 FOR N=1 TO 50
45 RANDOMIZE 65536*RND
50 LET D=(0.8-(INT (N/4))/5)/L
EV-LEV/25: IF D>0 AND N<12 THEN
GO TO 60
51 IF LEV=1 THEN LET D=0.1
52 IF LEV=2 THEN LET D=.06
53 IF LEV=3 THEN LET D=.01
60 LET A(N)=INT (RND*4)+1
70 FOR R=1 TO N
80 LET RT=64: GO SUB 9000+100*
A(R)
90 BEEP D, (PEEK 31018)/2-10
100 LET RT=0: GO SUB 9000+100*A
(R)
110 NEXT R
115 IF ERR=1 THEN RETURN
117 REM SEQUENZA GIOCATORE
120 FOR R=1 TO N
130 IF INKEY$="" THEN GO TO 13
0
140 PAUSE 0
150 LET K=CODE INKEY$
160 LET RT=64: GO SUB 8000+K
170 IF ANS<>A(R) THEN GO TO 70
00
180 BEEP D, (PEEK 31018)/2-10
190 LET RT=0: GO SUB 8000+K
200 NEXT R: PRINT FLASH 1: AT 1
6,23: N+1: PAUSE 25
210 NEXT N
220 CLS : FOR N=1 TO 3: FOR M=5
0 TO -20 STEP -3.5: BEEP .1, M: N
EXT M: NEXT N
225 REM ROUTINE VITTORIA
230 BEEP 2,-10: PRINT AT 10,0: "
RAGAZZO, SEI ENTRATO NELLA LEG-
GENDA. O SEI UN MOSTRO O HAI BA-
RATO. COMPLIMENTI IN OGNI CASO. ";

```



FLASH 1; AT 20,0; "PER GIOCARRE PR  
EMI UN TASTO"

240 PAUSE 0: CLS : GO TO 8  
7000 REM ROUTINE ERRORE  
7010 BEEP 2,-10: CLS : PRINT AT  
5,12; FLASH 1;"ERRORE!"; FLASH 0  
"VUOI VEDERE LA SEQUENZA ESA  
TTA?"; AT 10,13;"(S/N)": PAUSE 0  
7020 IF INKEY#="S" OR INKEY#="s"  
THEN LET ERR=1: LET D=.1: LET  
RT=0: CLS : RESTORE 9080: GO SU  
B 9040: RESTORE : GO SUB 70  
7030 IF N-1>TOP THEN LET TOP=N-  
1

7040 CLS : PRINT AT 10,0;"HAI RI  
CORDATO ";N-1;" PASSAGGI.": AT 13  
0;"IL RECORD E":TOP; AT 20,0: F  
LASH 1;"PER GIOCARRE PREMI UN TAS  
TO.": PAUSE 0: RESTORE : CLS : G  
O TO 8

8100 LET ANS=3: GO SUB 9300: RET  
URN  
8101 LET ANS=1: GO SUB 9100: RET  
URN  
8104 LET ANS=4: GO SUB 9400: RET  
URN  
8117 LET ANS=2: GO SUB 9200: RET  
URN  
8999 REM CARICAMENTO CODICI LM E  
INIZIALIZZAZIONE SCHERMO

9000 FOR n=31002 TO 31027  
9010 READ a: POKE n,a: NEXT n  
9020 DATA 17,22,0,58,24,121,111,  
58,25,121,103,6,5,14,10,54,12,35  
13,32,250,25,5,32,244,201  
9040 PLOT 32,152: DRAW 184,0: DR  
AW 0,-120: DRAW -184,0: DRAW 0,1  
20: DRAW 30,23: DRAW 124,0: DRAW  
30,-23

9045 PLOT 35,150: DRAW 178,0: DR  
AW 0,-99: DRAW -178,0: DRAW 0,99  
9050 PLOT 228,0: DRAW -30,23: DR  
AW -68,0: PLOT 118,23: DRAW -68,  
0: DRAW -30,-23: DRAW 208,0  
9055 PLOT 130,32: DRAW 0,-20: PL  
OT 118,32: DRAW 0,-20  
9060 FOR I=1 TO 4: READ X,Y: PLO  
T X,Y

9070 DRAW 75,0: DRAW 0,-36: DRAW  
-75,0: DRAW 0,36  
9080 DATA 42,141,130,141,42,93,1  
30,93  
9090 NEXT I  
9095 PRINT AT 16,6;"DIFF. ": FLA  
SH 1;LEV; FLASH 0;" LEVEL "  
FLASH 1;"1"

9100 POKE 31000,133: POKE 31001,  
88: POKE 31018,48+RT: IF RT=0 TH  
EN GO SUB 9999  
9101 IF RT<>0 THEN GO TO 9999  
9200 POKE 31000,144: POKE 31001,  
88: POKE 31018,40+RT: IF RT=0 TH  
EN GO SUB 9999  
9201 IF RT<>0 THEN GO TO 9999

9300 POKE 31000,69: POKE 31001,8  
9: POKE 31018,16+RT: IF RT=0 THE  
N GO SUB 9999  
9301 IF RT<>0 THEN GO TO 9999  
9400 POKE 31000,88: POKE 31001,8  
9: POKE 31018,32+RT: IF RT=0 THE  
N GO SUB 9999  
9401 IF RT<>0 THEN GO TO 9999  
9500 RETURN  
9999 LET L=USR 31002: RETURN

## Slot-machine

```

3 REM *****
4 REM ***   SLOT MACHINE ***
5 REM ***           di ***
6 REM *** Paolo De Rosa ***
7 REM ***   16 K RAM ***
8 REM *** (GRAFICA) ***
9 REM *****
10 FOR a=144 TO 158: FOR b=0 T
0 7: READ c
20 POKE (USR CHR# a)+b,c
30 NEXT b: NEXT a: RESTORE
50 DATA 0,0,14,31,63,127,103,1
27
60 DATA 48,64,176,248,252,252,
254,254
70 DATA 127,127,63,63,31,15,3,
0
80 DATA 254,254,252,252,248,24
0,192,0
90 DATA 0,0,14,30,55,125,111,1
25
100 DATA 48,64,176,184,108,252,
190,246
110 DATA 123,63,54,31,13,6,3,0
120 DATA 254,182,252,220,248,17
6,192,0
130 DATA 0,0,6,15,31,31,63,63
140 DATA 48,64,160,240,248,248,
252,252
150 DATA 63,63,31,31,15,7,3,0
160 DATA 252,252,248,248,240,22
4,192,0
170 DATA 0,0,14,31,31,63,39,127
180 DATA 0,255,246,223,245,111,
58,31
190 DATA 0,255,189,239,123,238,
188,248
2000 LOAD "SLOT MACH."
2100 SAVE "slot graf." LINE 10
2200 STOP
    
```

```

100 CLS : LET doll=25
200 BORDER 0: PAPER 0: INK 7: C
LS
220 POKE 23609,160
250 RANDOMIZE
270 FOR s=5 TO 21 STEP 8
    
```



```

280 PRINT PAPER 7; INK 0; AT 14
, s; "BAR "; AT 15, s; "BAR "; AT 16, s
; "BAR ": NEXT s
290 PRINT PAPER 6; INK 1; AT 0,
10; "SLOT-MACHINE"
300 PLOT 17, 32; DRAW 206, 0; DRA
W 0, 44; PLOT 226, 76; DRAW -210, 0
: DRAW 0, -44; PLOT 226, 76; DRAW
0, 95; DRAW -50, 0; PLOT 79, 171; D
RAW -65, 0; DRAW 0, -95; DRAW 3, 0
310 FOR n=112 TO 134 STEP 2: PL
OT 240, n; DRAW -6, 0; NEXT n
312 FOR n=8 TO 14
315 PRINT INK 7; AT n, 29; CHR$ 1
33; NEXT n
316 PRINT INK 7; AT 14, 28; CHR$
140+CHR$ 141
317 PRINT AT 18, 2; CHR$ 146+CHR$
147; AT 18, 26; CHR$ 146+CHR$ 147
320 PRINT BRIGHT 1; PAPER 7; I
NK 2; AT 1, 2; CHR$ 148+CHR$ 149+"
=2x"; AT 2, 2; CHR$ 150+CHR$
151+"
325 PRINT PAPER 7; INK 2; AT 3,
2; CHR$ 144+CHR$ 145+" =3x"
; AT 4, 2; CHR$ 146+CHR$ 147+"
330 PRINT PAPER 7; BRIGHT 1; I
NK 2; AT 5, 2; CHR$ 148+CHR$ 149+"
"+CHR$ 148+CHR$ 149+" =5x"; AT
6, 2; CHR$ 150+CHR$ 151+" "+CHR$
150+CHR$ 151+"
335 PRINT PAPER 7; INK 2; AT 7,
2; CHR$ 144+CHR$ 145+" "+CHR$ 144
+CHR$ 145+" =10x"; AT 8, 2; CHR$
146+CHR$ 147+" "+CHR$ 146+CHR$ 1
47+"
340 LET j$=CHR$ 152+CHR$ 153: L
ET k$=CHR$ 154+CHR$ 155+" ": PR
INT PAPER 7; BRIGHT 1; INK 1; AT
9, 2; j$+" "+j$+" "+j$+"=25x"; AT 1
0, 2; k$+k$+k$+" ": AT 11, 2; "gli
altri ="
345 LET f$=CHR$ 148+CHR$ 149+"
": LET g$=CHR$ 150+CHR$ 151+" ":
PRINT PAPER 7; INK 2; AT 1, 14; f
$+f$+f$+"=50 x"; AT 2, 14; g$+g$+g$
+"
350 LET b$=CHR$ 144+CHR$ 145+"
": LET d$=CHR$ 146+CHR$ 147+" ":
PRINT PAPER 7; BRIGHT 1; INK 2
; AT 3, 14; b$+b$+b$+"=100x"; AT 4, 1
4; d$+d$+d$+"
355 PRINT PAPER 7; INK 0; AT 6,
14; " 250"; AT 5, 14; "BAR
BAR BAR =x"
360 PRINT PAPER 7; BRIGHT 1; I
NK 0; AT 7, 14; "BAR BAR BAR =x"; AT
8, 14; "BAR BAR BAR500"
365 PRINT PAPER 7; INK 0; AT 9,
14; "BAR BAR BAR =x"; AT 10, 14; "BA
R BAR BAR 10"; AT 11, 14; "BAR BAR
BAR 00"
395 LET k=6: IF doll<6 THEN LE
T k=doll
400 PRINT PAPER 2; INK 7; AT 21

```

```

, 0; "HAI "; AT 21, 4; doll; " $
": AT 20, 3; "QUANTO PUNTI ? (MAX
": k; " $)
403 INPUT z$
404 IF z$="" THEN GO TO 403
405 LET z$=z$(1 TO 1)
410 IF CODE z$<49 OR CODE z$>54
THEN GO TO 403
420 LET z=VAL z$: LET z=INT z
430 IF z>doll THEN GO TO 400
470 PRINT PAPER 2; INK 7; AT 21
, 15; " HAI PUNTATO "; AT 21, 29; z;
$
475 PRINT AT 20, 3; "
"
476 LET doll=doll-z
477 PRINT PAPER 2; INK 7; AT 21
, 4; " ": AT 21, 4; doll; " $
"
480 PRINT AT 20, 5; INK 7; PAPER
2; "PREMI < M > PER TIRARE"
482 IF INKEY$="m" OR INKEY$="M"
THEN GO TO 484
483 GO TO 482
484 PRINT AT 20, 5; "
"
485 LET t=0: LET cont=0: LET w=
5
490 GO TO 490+INT (RND*69)+1
500 BEEP .04, 20: PRINT PAPER 7
; INK 0; AT 14, w; "BAR "; AT 15, w; "
BAR "; AT 16, w; "BAR ": FOR n=0 TO
t: NEXT n: GO SUB 1000
510 BEEP .04, 20: PRINT PAPER 7
; INK 2; AT 14, w; " ": AT 15, w; "
"+CHR$ 157+CHR$ 158+" ": AT 16, w
; PAPER 7; " ": FOR n=0 TO t:
NEXT n: GO SUB 1000
530 BEEP .04, 20: PRINT PAPER 7
; INK 0; AT 14, w; "BAR "; AT 15, w; "
": AT 16, w; "BAR ": FOR n=0 TO
t: NEXT n: GO SUB 1000
540 BEEP .04, 20: PRINT PAPER 7
; INK 2; AT 14, w; " ": AT 15, w; "
"+b$; AT 16, w; " "+d$; FOR n=0 TO
t: NEXT n: GO SUB 1000
545 BEEP .04, 20: PRINT PAPER 7
; INK 4; AT 14, w; " ": AT 15, w; "
"+CHR$ 156+CHR$ 145+" ": AT 16, w
; " "+CHR$ 146+CHR$ 147+" ": FOR
n=0 TO t: NEXT n: GO SUB 1000
550 BEEP .04, 20: PRINT PAPER 7
; INK 2; AT 14, w; " ": AT 15, w; "
"+CHR$ 148+CHR$ 149+" ": AT 16, w
; " "+CHR$ 150+CHR$ 151+" ": FOR
n=0 TO t: NEXT n: GO SUB 1000
560 BEEP .04, 20: PRINT PAPER 7
; INK 0; AT 14, w; " ": AT 15, w; "
BAR "; AT 16, w; " ": FOR n=0 TO
t: NEXT n: GO SUB 1000
570 BEEP .04, 20: PRINT PAPER 7
; INK 1; AT 14, w; " ": AT 15, w; "
"+CHR$ 152+CHR$ 153+" ": AT 16, w
; " "+CHR$ 154+CHR$ 155+" ": FOR
n=0 TO t: NEXT n: GO SUB 1000
900 GO TO 500

```





```

5100 PRINT AT 20,0;"
      ";AT 21,0;" PAPER 7; IN
K 2;" PREMI < Z > PER UN'ALTRO
TIRO"
5120 IF INKEY$="z" OR INKEY$="Z"
THEN PRINT AT 21,0;"
      ": GO TO 3
20
5130 GO TO 5110
5200 PRINT AT 19,0;"HAI PERSO TU
TTI I TUOI SOLDI !!!PREMI S PER
UN ALTRA PARTITA PREMI N PER
SMETTERE
      "
5210 IF INKEY$="S" OR INKEY$="s"
THEN GO TO 30
5220 IF INKEY$="N" OR INKEY$="n"
THEN STOP
5230 GO TO 5210
6000 CLEAR
6010 SAVE "SLOT MACH." LINE 100
6020 STOP

```

```

120 PRINT "SIGNIFICATO DEI SEG
NI:";"q$+" ----> POSIZIONE DELLE
DITA""
      " SULLA TASTIERA""
q$'r$+" ----> BARRE'r$+" ---->(
TUTTE LE CORDE PREMUTE""
CON LA PRESSIONE DI UN'q$+" ----
> SOLO DITO)"
130 PRINT #0;"X PER CONTINUARE
C COPY M MENU"
140 IF INKEY$="" THEN GO TO 14
0
141 IF INKEY$="M" THEN GO TO 1
5
142 IF INKEY$="C" THEN COPY :
GO TO 140
143 IF INKEY$="X" THEN GO TO 1
45
144 GO TO 140
145 GO SUB 395
150 PRINT AT 10,0;"NOMI E NUMER
I DELLE CORDE"
160 GO SUB 1000
170 PRINT #0;"V PER RIVEDERE M
PER MENU C COPY"
180 IF INKEY$="" THEN GO TO 18
0
185 IF INKEY$="C" THEN COPY :
GO TO 180
190 IF INKEY$="M" THEN GO TO 1
5
195 IF INKEY$="V" THEN GO TO 1
00
196 GO TO 180
200 LET FLAG=0: GO SUB 2500
205 PRINT AT 8,0;"##### GIR
I ARMONICI #####"
210 PRINT AT 0,0;"1)-> DOMAG"
"2)-> REBMAG""3)-> REMAG""4)->
MIBMAG""5)-> MIMAG""6)-> FAMA
G""7)-> SOLBMAG""8)-> SOLMAG"
"9)-> LABMAG""10)->LAMAG""11)-
>SIBMAG""12)->SIMAG"
211 INPUT Z$: LET FLAG=1: LET Z
=VAL Z$: IF Z<1 OR Z>12 THEN GO
TO 211
212 FOR F=10 TO 21: PRINT AT F,
0;W$: NEXT F
215 IF Z=1 THEN RESTORE 2000:
READ A$,B$: LET Z$="DOMAG": PRIN
T AT 9,0;"1o ACCORDO GIRO DI D
O MAG."; GO SUB 520: RESTORE 205
5: READ A$,B$: LET Z$="LAMIN": P
RINT AT 9,0;"2": GO SUB 520: RES
TORE 2012: READ A$,B$: LET Z$="R
EMIN": PRINT AT 9,0;"3": GO SUB
520: RESTORE 2045: READ A$,B$: L
ET Z$="SOL7": PRINT AT 9,0;"4":
GO SUB 520
216 IF Z=2 THEN RESTORE 2006:
READ A$,B$: LET Z$="REBMAG": PRI
NT AT 9,0;"1o ACCORDO GIRO DI
REB MAG.": GO SUB 520: RESTORE 2
061: READ A$,B$: LET Z$="SIBMIN"
: PRINT AT 9,0;"2": GO SUB 520:
RESTORE 2019: READ A$,B$: LET Z$
="MIBMIN": PRINT AT 9,0;"3": GO

```

## Accordi per la chitarra

0)REM

```

*****
* SCRITTO E REALIZZATO DA*
*
* DONATI MARIO
* VIA CLAUDIO ASELLO 74 *
* 00175 ROMA *
* TEL. 06/7660227 *
*****

```

```

10 FOR f=USR "a" TO USR "b"+7:
READ a: POKE f,a: NEXT f: DATA
60,126,255,255,255,255,126,60,24
,24,24,24,24,24,24,24
15 LET W$=""

```

```

      ": LET FLAG=0
20 GO SUB 2500
30 POKE 23658,8
40 GO SUB 2520
45 LET q$=CHR$: 144: LET r$=CHR
$ 145
50 PRINT AT 8,0;"#####
MENU#####
60 PRINT "1)----> ISTRUZIONI"
"2)----> GIRI ARMONICI""3)----> A
CCORDATURA""4)----> PREPARAZIONE
""5)----> ACCORDI"
70 INPUT "----> ";0$
75 LET O=VAL 0$
80 GO TO (100 AND O=1)+(200 AN
D O=2)+(300 AND O=3)+(400 AND O=
4)+(500 AND O=5)+(70 AND O>5)+(7
0 AND O<=0)
100 GO SUB 2500
110 PRINT AT 8,0;"##### IS
TRUZIONI #####"

```



SUB 520: RESTORE 2051: READ A\$,B\$:  
\$: LET Z\$="LAB7": PRINT AT 9,0;"  
4": GO SUB 520

217 IF Z=3 THEN RESTORE 2011:  
READ A\$,B\$: LET Z\$="REMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI R  
E MAG.": GO SUB 520: RESTORE 206  
7: READ A\$,B\$: LET Z\$="SIMIN": P  
RINT AT 9,0;"2": GO SUB 520: RES  
TORE 2025: READ A\$,B\$: LET Z\$="M  
IMIN": PRINT AT 9,0;"3": GO SUB  
520: RESTORE 2057: READ A\$,B\$: L  
ET Z\$="LA7": PRINT AT 9,0;"4": G  
O SUB 520

218 IF Z=4 THEN RESTORE 2018:  
READ A\$,B\$: LET Z\$="MIBMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI  
MIB MAG.": GO SUB 520: RESTORE 2  
001: READ A\$,B\$: LET Z\$="DOMIN":  
PRINT AT 9,0;"2": GO SUB 520: R  
ESTORE 2031: READ A\$,B\$: LET Z\$=  
"FAMIN": PRINT AT 9,0;"3": GO SU  
B 520: RESTORE 2063: READ A\$,B\$:  
LET Z\$="SIB7": PRINT AT 9,0;"4"  
: GO SUB 520

219 IF Z=5 THEN RESTORE 2024:  
READ A\$,B\$: LET Z\$="MIMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI M  
IMAG.": GO SUB 520: RESTORE 2089  
: READ A\$,B\$: LET Z\$="DOMMIN": P  
RINT AT 9,0;"2": GO SUB 520: RES  
TORE 2073: READ A\$,B\$: LET Z\$="F  
A#MIN": PRINT AT 9,0;"3": GO SUB  
520: RESTORE 2069: READ A\$,B\$:  
LET Z\$="SI7": PRINT AT 9,0;"4":  
GO SUB 520

220 IF Z=6 THEN RESTORE 2030:  
READ A\$,B\$: LET Z\$="FAMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI F  
A MAG.": GO SUB 520: RESTORE 201  
2: READ A\$,B\$: LET Z\$="REMIN": P  
RINT AT 9,0;"2": GO SUB 520: RES  
TORE 2043: READ A\$,B\$: LET Z\$="S  
OLMIN": PRINT AT 9,0;"3": GO SUB  
520: RESTORE 2003: READ A\$,B\$:  
LET Z\$="DO7": PRINT AT 9,0;"4":  
GO SUB 520

221 IF Z=7 THEN RESTORE 2037:  
READ A\$,B\$: LET Z\$="SOLBMAG": PR  
INT AT 9,0;"1o ACCORDO GIRO DI  
SOLB MAG.": GO SUB 520: RESTORE  
2019: READ A\$,B\$: LET Z\$="MIBMI  
N": PRINT AT 9,0;"2": GO SUB 520  
: RESTORE 2049: READ A\$,B\$: LET  
Z\$="LABMIN": PRINT AT 9,0;"3": G  
O SUB 520: RESTORE 2009: READ A\$,  
B\$: LET Z\$="REB7": PRINT AT 9,0  
;"4": GO SUB 520

222 IF Z=8 THEN RESTORE 2042:  
READ A\$,B\$: LET Z\$="SOLMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI  
SOL MAG.": GO SUB 520: RESTORE 2  
025: READ A\$,B\$: LET Z\$="MIMIN":  
PRINT AT 9,0;"2": GO SUB 520: R  
ESTORE 2055: READ A\$,B\$: LET Z\$=  
"LAMIN": PRINT AT 9,0;"3": GO SU

B 520: RESTORE 2014: READ A\$,B\$:  
LET Z\$="RE7": PRINT AT 9,0;"4":  
GO SUB 520

223 IF Z=9 THEN RESTORE 2048:  
READ A\$,B\$: LET Z\$="LABMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI  
LAB MAG.": GO SUB 520: RESTORE 2  
031: READ A\$,B\$: LET Z\$="FAMIN":  
PRINT AT 9,0;"2": GO SUB 520: R  
ESTORE 2061: READ A\$,B\$: LET Z\$=  
"SIBMIN": PRINT AT 9,0;"3": GO S  
UB 520: RESTORE 2021: READ A\$,B\$:  
LET Z\$="MIB7": PRINT AT 9,0;"4  
": GO SUB 520

224 IF Z=10 THEN RESTORE 2054:  
READ A\$,B\$: LET Z\$="LAMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI  
LA MAG.": GO SUB 520: RESTORE 20  
73: READ A\$,B\$: LET Z\$="FA#MIN":  
PRINT AT 9,0;"2": GO SUB 520: R  
ESTORE 2067: READ A\$,B\$: LET Z\$=  
"SIMIN": PRINT AT 9,0;"3": GO SU  
B 520: RESTORE 2027: READ A\$,B\$:  
LET Z\$="MI7": PRINT AT 9,0;"4":  
GO SUB 520

225 IF Z=11 THEN RESTORE 2060:  
READ A\$,B\$: LET Z\$="SIBMAG": PR  
INT AT 9,0;"1o ACCORDO GIRO DI  
SIB MAG.": GO SUB 520: RESTORE  
2043: READ A\$,B\$: LET Z\$="SOLMIN  
": PRINT AT 9,0;"2": GO SUB 520:  
RESTORE 2001: READ A\$,B\$: LET Z  
\$="DOMIN": PRINT AT 9,0;"3": GO  
SUB 520: RESTORE 2033: READ A\$,B  
\$: LET Z\$="FA7": PRINT AT 9,0;"4  
": GO SUB 520

226 IF Z=12 THEN RESTORE 2066:  
READ A\$,B\$: LET Z\$="SIMAG": PRIN  
T AT 9,0;"1o ACCORDO GIRO DI  
SI MAG.": GO SUB 520: RESTORE 20  
90: READ A\$,B\$: LET Z\$="SOL#MIN"  
: PRINT AT 9,0;"2": GO SUB 520:  
RESTORE 2089: READ A\$,B\$: LET Z \$  
="DOMMIN": PRINT AT 9,0;"3": GO  
SUB 520: RESTORE 2075: READ A\$,B  
\$: LET Z\$="FA#7": PRINT AT 9,0;"  
4": GO SUB 520

250 GO TO 200  
300 GO SUB 2500  
310 PRINT AT 8,0;"@@@@@@@@ ACC  
ORDATURA @@@@@@@@@@

320 PRINT AT 10,0;"LE FASI PER  
L'ACCORDATURA DELLA""CHITARRA S  
ONO 6:"

330 PRINT ""1)ACCORDARE LA 5a C  
ORDA""(LA) A CORISTA""PIU' O  
MENO DOVRETE RAGGIUNGERE""QUES  
TA TONALITA'": PAUSE 50: BEEP 2,  
-17

331 PRINT #0;AT 0,0;"UN TASTO P  
ER CONTINUARE C COPY"

332 IF INKEY\$="" THEN GO TO 33  
1

333 IF INKEY\$="C" THEN COPY  
334 INPUT "": GO SUB 395: IF FL  
AG=1 THEN RETURN





```

335 PRINT AT 10,0;"2)PREMENDO I
L 5o TASTO SULLA 5a""CORDA OTTE
RREMO IL RE (4a CORDA)": BEEP 2,
-12
340 PAUSE 50: PRINT ""3)PREMEND
O IL 5o TASTO SULLA 4a""CORDA O
TTERREMO IL SOL(3a CORDA)": BEEP
2,-7
345 PAUSE 50: PRINT ""4)PREMEND
O IL 4o TASTO SULLA 5a""CORDA O
TTERREMO IL SI (2a CORDA)": BEEP
2,-3
350 PAUSE 50: PRINT ""5)PREMEND
O IL 5o TASTO SULLA 2a""CORDA O
TTERREMO IL MIc(1a CORDA)": BEEP
2,2
355 LET FLAG=1: GO SUB 331
360 PRINT AT 10,0;"ADESSO ACCOR
DIAMO IL""MI BASSO (MIb) CON IL
""MI CANTINO (MIc).""DOVRETE
RAGGIUNGERE QUESTA""TONALITA':
": PAUSE 100: BEEP 2,-22
365 PRINT #0: AT 0,0;"PREMI M PE
R MENU'S PER RISENTIRE""V PER R
IVEDERE C PER COPY": PAUSE 0
366 IF INKEY$="S" THEN BEEP 1,
-22: BEEP 1,-17: BEEP 1,-12: BEE
P 1,-7: BEEP 1,-3: BEEP 1,2: GO
TO 365
367 IF INKEY$="M" THEN GO TO 1
5
368 IF INKEY$="V" THEN LET FLA
G=0: GO TO 300
369 IF INKEY$="C" THEN COPY :
GO TO 365
370 GO TO 366
395 FOR F=9 TO 21: PRINT AT F,0
;W$: NEXT F: RETURN
400 GO SUB 2500: GO SUB 1000
405 PRINT AT 8,0;"@@@@@@@@ PRE
PARAZIONE @@@@@@@@@@
410 PRINT AT 10,0;"QUESTO GRAFI
CO RAPPRESENTA UNA""PARTE DI TA
STIERA DELLA CHITARRA": GO SUB 4
89
415 PRINT AT 10,0;"LE LINEE LON
GITUDINALI RAPPRE""SENTANO LE
6 CORDE NEL SEGUENTE""ORDINE":
GO SUB 489
420 PRINT AT 10,0;"1)->MI CANTI
NO 2)->SI 3)->SOL ""4)->RE 5)->
LA 6)->MI BASSO ""W$: GO SUB
B 489
425 PRINT AT 10,0;"LE LINEE TRA
SVERSALI RAPPRES""TANO I TAS
TI. ""GO SUB
489
430 PRINT AT 10,0;"IL CAPOTASTO
VIENE DETERMINATO""DALLA POSIZ
IONE IN CUI SI TROVA""IL DITO I
NDICE DELLA MANO""SINISTRA. ESE
MPIO:": GO SUB 489
435 PRINT AT 10,0;"SE IL DITO I
NDICE DELLA MANO SX""SI TROVA S
UL PRIMO TASTO, STIAMO""SUONANDO
ALL'ALTEZZA DEL ""PRIMO

```

```

CAPOTASTO ""GO S
UB 489
445 PRINT AT 10,0;"SE SI TROVA
SUL SECONDO TASTO, ""STIAMO SU
ONANDO ALL'ALTEZZA DEL ""SECONDO
O CAPOTASTO ECC... ""W$:
GO SUB 489
450 PRINT AT 10,0;"I NUMERI IN
BASSO INDICANO ""IL CAPOTA
STO. ""W$:W$:
GO SUB 489
452 FOR F=10 TO 21: PRINT AT F,
0;W$: NEXT F
455 PRINT AT 10,0;"NEL PROGRAMM
A SONO MEMORIZZATI ""90 ACCORDI.
E' IMPORTANTE""RICORDARE CHE I
NOMI DELLE NOTE""VANNO INSERITI
CON IL SEGUENTE""SISTEMA: ""
""DO MAGGIORE = ~DOMAG~""DO MINO
RE = ~DOMIN~""DO SESTA = ~DO6~
""DO SETTIMA = ~DO7~""DO SETTIM
A ECCEDEnte = ~DO7+~""DO MINORE
SETTIMA = ~DOMIN7~
456 PRINT #0: AT 0,0;W$: GO SUB
489
460 PRINT AT 10,0;"L'ESEMPIO PR
ECEDENTE VALE PER ""TUTTE LE
NOTE. INOLTRE BISOGNA ""RICORD
ARSI CHE NON SI PUD ""(PE
R ES.) INTRODURRE ~FA~
INTENDENDO ~FA MAGGIORE~
""ALTRIMENTI IL PROGRAMMA NON
""RICONOSCE LA NOTA
""W$:W$: "N.B.--> NON METTE
RE SPAZI QUANDO""SI SCRIVE LA N
OTA... ""W$
465 PRINT #0: AT 0,0;"PREMI M PE
R MENU' V PER RIVEDERE""C PER C
OPY"
470 IF INKEY$="M" THEN GO TO 1
5
475 IF INKEY$="V" THEN GO TO 4
00
480 IF INKEY$="C" THEN COPY :
GO TO 465
485 GO TO 470
489 PRINT #0;"C COPY M MENU'X C
ONTINUA ""PAUSE 0: INPUT "
490 IF INKEY$="M" THEN GO TO 1
5
491 IF INKEY$="C" THEN COPY :
GO TO 495
492 IF INKEY$<>"X" THEN GO SUB
1077: GO TO 489
494 IF INKEY$="" THEN GO TO 49
0
495 GO SUB 1077: RETURN
500 POKE 23658,8: GO SUB 2500
501 PRINT AT 8,0;"##### A
CCORDI #####
502 IF FLAG=1 THEN RETURN
503 INPUT "Accordo? ";z$: IF z$
="" THEN BEEP .2,-20: GO TO 500
505 IF z$(1 TO 2)<>"DO" AND z$(
1 TO 2)<>"RE" AND z$(1 TO 2)<>"M

```



```

I" AND z$(1 TO 2)<>"FA" AND z$(1
2) <>"GO" AND z$(1 TO 2)<>"SI
" AND z$(1 TO 2)<>"LA" THEN PRI
NT AT 10,3; FLASH 1;"NON ESISTE
QUESTO ACCORDO"; FLASH 0; BEEP 1
,-10: PRINT AT 10,0;W$: GO TO 50

```

```

520 FOR F=1 TO 91: RESTORE 1999
+F: READ a$,B$: IF A$(17 TO )=Z$
THEN GO TO 530
525 NEXT F

```

```

527 PRINT AT 10,3; FLASH 1;"NON
CONOSCO QUESTO ACCORDO"; FLASH
0; BEEP 1,0: PRINT AT 10,0;W$: G
O TO 502

```

```

530 LET a=VAL a$(1 TO 2): LET b
=VAL a$(3 TO 4): LET c=VAL a$(5
TO 6): LET d=VAL a$(7 TO 8)

```

```

540 LET aa=VAL a$(9 TO 10): LET
bb=VAL a$(11 TO 12): LET cc=VAL
a$(13 TO 14): LET dd=VAL a$(15
TO 16): LET a$=a$(17 TO )

```

```

545 GO SUB 1000
550 PRINT AT A,B;q$;AT C,D;q$;A
T AA,BB;q$;AT CC,DD;q$;AT 10,7;"
ACCORDO ";A$

```

```

555 IF B$(1)="1" THEN LET P=VA
L B$(2 TO 3): LET O=VAL B$(4 TO
5): LET I=VAL B$(6 TO 7): LET U=
VAL B$(8 TO 9): LET II=VAL B$(10
TO 11): LET UU=VAL B$(12 TO 13)
: LET PP=VAL B$(14 TO 15): LET O
=VAL B$(16 TO 17): PRINT AT P,0
;q$;AT I,U;r$;AT II,UU;r$;AT PP,
00;q$

```

```

560 PRINT #0; " X CONTINUA C
OPY M MENU " "
570 IF INKEY$="" THEN GO TO 57
0

```

```

575 IF INKEY$="C" THEN COPY :
GO TO 570

```

```

580 IF INKEY$="M" THEN GO TO 1
5

```

```

585 IF INKEY$="X" THEN FOR F=1
0 TO 21: PRINT AT F,0;W$: NEXT F
: GO TO 502

```

```

587 IF INKEY$="M" THEN GO TO 2
0

```

```

590 GO TO 570

```

```

1000 FOR f=0 TO 3: PLOT 0,f: DRA
W 200,0: NEXT f

```

```

1010 FOR f=11 TO 51 STEP 8: PLOT
0,f: DRAW 200,0: NEXT f

```

```

1065 FOR f=59 TO 63: PLOT 0,f: D
RAW 200,0: NEXT f

```

```

1070 FOR f=0 TO 200 STEP 30: PLO
T f,0: DRAW 0,63: NEXT f

```

```

1075 FOR F=15 TO 20: PRINT AT F,
25; ("1-M1c" AND F=15)+("2-SI" AN
D F=16)+("3-SOL" AND F=17)+("4-R
E" AND F=18)+("5-LA" AND F=19)+("
6-M1b" AND F=20): NEXT F

```

```

1077 PRINT #0;AT 0,0;" 1 2 3
4 5 6"

```

```

1080 RETURN
2000 DATA "1602000018051909DOMAG

```

```

" 0"

```

```

2001 DATA "1509161317161816DOMIN
" 0"

```

```

2002 DATA "1602000018050000D06",
"0"

```

```

2003 DATA "1602180519091709D07",
"0"

```

```

2004 DATA "0000000018061909D07+"
"0"

```

```

2005 DATA "1509000017090000DOMIN
7","11502170218022002"

```

```

2006 DATA "1605000018091912REBMA
G","11502170218022002"

```

```

2007 DATA "1605170218050000REBMI
N","0"

```

```

2008 DATA "1605170918091512REB6"
"0"

```

```

2009 DATA "1605180915121712REB7"
"0"

```

```

2010 DATA "1512171200000000REBMI
N7","11505170518052005"

```

```

2011 DATA "1505170516090000REMAG
" 0"

```

```

2012 DATA "1502170516090000REM IN
" 0"

```

```

2013 DATA "1505170519051609RE6",
"0"

```

```

2014 DATA "1602150517050000RE7",
"0"

```

```

2015 DATA "0000000000000000RE7+"
"11505170518052005"

```

```

2016 DATA "1502160217050000REM IN
7","0"

```

```

2017 DATA "1809191200000000REB7+
" "11502170218022002"

```

```

2018 DATA "1612181600000000MIBMA
G","11509170918092009"

```

```

2019 DATA "1505170916121812MIBMI
N","0"

```

```

2020 DATA "1509170900000000MIB6"
"11502170218022002"

```

```

2021 DATA "1802170916051509MIB7"
"0"

```

```

2022 DATA "1802150916091709MIB7+
" 0"

```

```

2023 DATA "1802150516051709MIBMI
N7","0"

```

```

2024 DATA "1702180519050000MIMAG
" 0"

```

```

2025 DATA "1805190500000000MIMIN
" 0"

```

```

2026 DATA "1512171200000000M16",
"11505170518052005"

```

```

2027 DATA "1702180519051609M17",
"0"

```

```

2028 DATA "1702180219050000M17+"
"0"

```

```

2029 DATA "1905000000000000MIMIN
7","0"

```

```

2030 DATA "0000190917051809FAMAG
" "11502170218022002"

```

```

2031 DATA "1809190900000000FAMIN
" "11502170218022002"

```

```

2032 DATA "1502160217050000FA6",
"0"

```

```

2033 DATA "0000161217051809FA7"

```



"11502170218022002"  
2034 DATA "16021705180900000FA7+  
"0"  
2035 DATA "190900000000000000FAMIN  
7" "11502170218022002"  
2036 DATA "1505160517051812SOLBM  
IN" "0"  
2037 DATA "1505160517091812SOLBM  
AG" "0"  
2038 DATA "1505160517091802SOLB6  
" "0"  
2039 DATA "1605170918120000SOLB7  
" "0"  
2040 DATA "1502160517091812SOLB7  
" "0"  
2041 DATA "000000000000000000SOLBM  
IN7" "11505170518052005"  
2042 DATA "15091905200900000SOLMA  
G" "0"  
2043 DATA "181619160000000000SOLMI  
N" "11509170918092009"  
2044 DATA "19050000000000000SOL6"  
"0"  
2045 DATA "15021905200900000SOL7"  
"0"  
2046 DATA "1505160917121816SOL7+  
" "0"  
2047 DATA "000000000000000000SOLMI  
N7" "11509170918092009"  
2048 DATA "1512161217161820LABMA  
G" "0"  
2049 DATA "182000000000000000LABMI  
N" "11512171218122012"  
2050 DATA "000000000000000000LAB6"  
"11502170218022002"  
2051 DATA "150500000000000000LAB7"  
"11502170218022002"  
2052 DATA "000000000000000000LABMI  
N7" "11512171218122012"  
2053 DATA "150900000000000000LAB7+  
" "11512171218122012"  
2054 DATA "1605170518050000LAMAG  
" "0"  
2055 DATA "1602170518050000LAMIN  
" "0"  
2056 DATA "000000000000000000LA6"  
"11505170518052005"  
2057 DATA "150900000000000000LA7"  
"11505170518052005"  
2058 DATA "151200000000000000LA7+  
" "11505170518052005"  
2059 DATA "1602170518051509LAMIN  
7" "0"  
2060 DATA "16091709180900000SIBMA  
G" "11502170218022002"  
2061 DATA "16051709180900000SIBMI  
N" "11502170218022002"  
2062 DATA "190200000000000000SIB6"  
"11509170918092009"  
2063 DATA "151200000000000000SIB7"  
"11509170918092009"  
2064 DATA "17051609180900000SIB7+  
" "11502170218022002"  
2065 DATA "1605170918091512SIBMI  
N7" "0"  
2066 DATA "1612171218120000SIMGAG

"11505170518052005"  
2067 DATA "16091712181200000SIMIN  
" "11505170518052005"  
2068 DATA "190500000000000000SI6"  
"11512171218122012"  
2069 DATA "1802150517051905SI7"  
"0"  
2070 DATA "1709161218120000SI7+  
" "11505170518052005"  
2071 DATA "160900000000000000SIMIN  
7" "11505170518052005"  
2072 DATA "1709181219120000FA#MA  
G" "11505170518052005"  
2073 DATA "1812191200000000FA#MI  
N" "11505170518052005"  
2074 DATA "1502160917051809FA#6"  
"0"  
2075 DATA "1709191216160000FA#7"  
"11505170518052005"  
2076 DATA "191200000000000000FA#MI  
N7" "11502170218022002"  
2077 DATA "1502160517091812FA#7+  
" "0"  
2078 DATA "1602180200000000DOMIN  
6" "0"  
2079 DATA "1702180219052005REBMI  
N6" "0"  
2080 DATA "1502170500000000REMIN  
6" "0"  
2081 DATA "1505170900000000MIBMI  
N6" "11502170218022002"  
2082 DATA "1509171200000000MIMIN  
6" "11502170218022002"  
2083 DATA "1502160217020000FAMIN  
6" "0"  
2084 DATA "1509160917090000SOLMI  
N6" "11505170518052005"  
2085 DATA "1502170218020000LABMI  
N6" "0"  
2086 DATA "1602150517051805LAMIN  
6" "0"  
2087 DATA "1605150917091809SIMIN  
6" "0"  
2088 DATA "1702150519050000SIMIN  
6" "0"  
2089 DATA "1616172018200000D0#MI  
N" "11512171218122012"  
2090 DATA "172018200000000000SOL#M  
IN" "11512171218122012"  
2500 CLS : PRINT "#####"  
##### C  
H I T A R R A ""#####  
#####"  
2510 RETURN  
2520 FDR F=7 TO 21: PRINT AT F,0  
;W\$: NEXT F  
2530 RETURN

## Numeri in virgola mobile

1 LET q=PEEK 23637+256\*PEEK 2  
3638+5: GO SUB 200



```

2 REM //****//****//****//
10 CLS : PRINT AT 2,2;"CONVERS
IONE DEC-FP"; AT 5,2;"decimale"; A
T 7,2;"FP full"; AT 9,2;"FP compr
."
20 DIM f(5)
30 PRINT #1; AT 0,0;"numero Dec
- Full - Compr", "(D-F-C => ?":
PAUSE 0: LET x$=CHR$ PEEK 23556
40 IF x$="D" THEN INPUT "immet
tti il numero decimale", "=">"; d
: PRINT AT 5,13;d: GO TO 110
50 IF x$="C" THEN GO TO 70
55 IF x$<>"F" THEN GO TO 30
60 INPUT "immetti in decimale,
una per volta, le 5 cifre de
l numero full FP =>"; f(1); " "
;f(2); " "; f(3); " "; f(4); " "; f(5)
65 PRINT AT 7,13;f(1); " "; f(2)
; " "; f(3); " "; f(4); " "; f(5): GO
TO 120
70 PRINT AT 9,13;: FOR n=1 TO
5: INPUT "immetti in decimale, u
na per volta, le 2...5 cifre
del numero compr. FP, o solo ENTE
R quando e' completo => "; LINE
x$
75 IF x$="" THEN GO TO 80
77 PRINT x$; " "; POKE q+6+n,V
AL x$: NEXT n
80 IF n<3 OR PEEK (q+7)=192 TH
EN PRINT AT 20,0;"numero invali
do": GO TO 100
85 POKE q+6+n,51: POKE q+7+n,2
51-n: RANDOMIZE USR (q+5): GO SU
B 130
90 PRINT AT 5,13;: RANDOMIZE U
SR (q+14)
100 PRINT #1;"un tasto per cont
inuare", "o BREAK": PAUSE 0: RUN
110 RANDOMIZE d=USR q: GO SUB 1
30: GO SUB 140: GO TO 100
120 FOR n=1 TO 5: POKE 23702+n,
f(n): NEXT n: GO SUB 140: GO TO
90
130 PRINT AT 7,13;: FOR n=1 TO
5: PRINT PEEK (23702+n); " "; NE
XT n: RETURN
140 PRINT AT 9,13;: IF (PEEK 23
703<81 OR PEEK 23703>143) AND PE
EK 23707<>0 THEN PRINT "non con
sentito": RETURN
141 LET c=0: FOR n=5 TO 1 STEP
-1: LET f(n)=PEEK (23702+n)
142 IF c<>0 OR f(n)<>0 THEN LE
T c=c+1
143 NEXT n
144 IF c>1 THEN LET c=c-2
145 IF f(1)<80 THEN LET f(1)=f
(1)+256
146 IF f(1)>143 OR f(1)<81 THEN
PRINT 64*c; " "; f(1)-80;: LET x
=1: LET c=c+x: GO TO 150
147 PRINT f(1)-80+64*c; " "; f(2)

```

```

;: LET x=2
150 FOR n=1 TO c: PRINT " "; f(x
+n);: NEXT n: RETURN
200 RESTORE
210 FOR n=0 TO 19: READ a: POKE
q+n,a: NEXT n: RETURN
220 DATA 239,61,193,56,201,239,
52,0,0,0,0,0,0,0,239,225,56,195,
227,45

```

## Scroll orizzontale

```

1 LET iniz=PEEK 23637+256*PEE
K 23638+5: GO SUB 1000: GO TO 30
2 REM //****//****//****//
//****//****//****//****//
*****//****//****//****//
//****//****//****//****//
*****//
10 IF INKEY$<>" " THEN GO TO 1
0
20 GO SUB 200
30 PRINT AT 2,0;" La routine e
stata caricata. Digita il mes
saggio e dai ENTER."
40 INPUT a$
50 IF LEN a$<1 THEN GO TO 40
60 LET a$=""+a$
70 IF INKEY$<>" " THEN GO TO 7
0
90 PRINT AT 14,0;"premi R per
un nuovo RUN oppure E per
uscire."
100 GO SUB 200
110 IF INKEY$="R" OR INKEY$="r"
THEN RUN
120 IF INKEY$<>"q" AND INKEY$<>
"Q" THEN GO TO 100
130 STOP
200 FOR i=1 TO LEN a$
210 POKE iniz+i, CODE a$(i)
220 RANDOMIZE USR iniz
230 IF INKEY$<>" " THEN RETURN
240 NEXT i
250 GO TO 200
1000 RESTORE
1010 FOR q=0 TO 131: READ a: POK
E iniz+q,a: NEXT q: RETURN
1020 DATA 62,122,237,91,54,92,38
0,111,41,41,41,25,235,14,4,33,3
1,72,6,4,26,119,36,16,252,19,26,
6,4,119,36,16,252,19,124,214,8,1
03,62,32,133,111,13,32,229,6,8,1
97,33,127,79,118,1,9,128,0,16,25
3,13,32,250,229,126,203,23,43,6,
31
1030 DATA 203,22,43,16,251,225,3
7,124,254,71,32,237,198,8,103,12
5,214,32,111,48,228,33,127,79,11
8,1,9,128,0,16,253,13,32,250,229
,175,6,32,203,22,43,16,251,225,3
7,124,254,71,32,240,198,8,103,12
5,214,32,111,48,231,193,16,173
1040 DATA 201

```



Nel primo articolo un programma Pascal era stato descritto come una "scatola" contenente altre "sottoscatole"; questi moduli di livello inferiore si possono definire più propriamente procedure e funzioni.

Il significato di una procedura o di una funzione è simile a quello di una subroutine basic, serve cioè a realizzare delle operazioni che verranno utilizzate in diverse parti del programma su dati diversi.

Una procedura, e analogamente una funzione, possono venire richiamate solo dopo la loro definizione e solo all'interno del modulo che le contiene.

Una procedura è costituita da un header (fig. 1) e da un block (fig. 2) e può a sua volta contenere la definizione di altre procedure o funzioni, che non possono però essere utilizzate dal programma principale per la regola vista prima.

## Chiamata di procedura

A differenza delle subroutines basic, le procedure Pascal hanno una maggiore generalità di applicazione, perchè possono venire parametrizzate.

I parametri sono il mezzo attraverso il quale avviene lo scambio di informazione fra il modulo che chiama la procedura e la procedura chiamata (fig. 3). Durante la definizione di una procedura si definiscono le operazioni su parametri formali, il cui valore verrà poi definito al momento della chiamata dalla associazione posizionale con i parametri attuali.

Poichè lo scambio di informazioni può avvenire secondo modalità differenti, esistono diversi tipi di parametri.

## Parametri VALUE

Servono per passare informazioni dall'ambiente chiamante alla procedura: i parametri attuali vengono copiati localmente dalla procedura e solo il valore di questa copia viene modificato. Di conseguenza al termine della procedura il valore dei

# Un po' di PASCAL

di Monica Fumagalli

Fig. 1 - Header di procedura

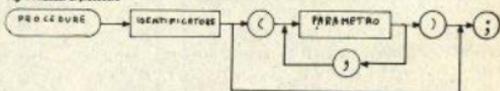


Fig. 2 - Block di procedura/funzione

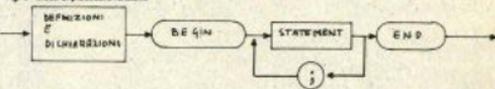


Fig. 3 - Chiamata di procedura

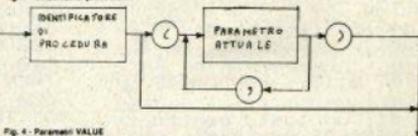


Fig. 4 - Parametri VALUE

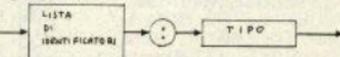


Fig. 5 - Parametri VARIABLE

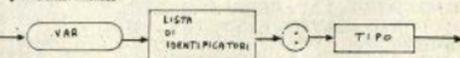


Fig. 6 - Header di funzione



Fig. 7 - Chiamata di funzione



EDIZIONE  
ITALIANA

# QLUB

**Notiziario QL**

**COMUNICATO PER TUTTI  
I POSSESSORI DI QL  
E PER QUANTI HANNO  
INTENZIONE A DIVENTARLO**

IN OTTOBRE È USCITO  
IL PRIMO NUMERO DI  
QLUB — NOTIZIARIO QL,  
RISERVATO IN ESCLUSIVA  
A TUTTI I POSSESSORI DEL QL  
CON GARANZIA ITALIANA.

SE STAI PER COMPRARE UN QL  
CONTROLLA CHE SIA DOTATO  
DELLA GARANZIA ITALIANA:  
È L'UNICO MEZZO  
PER RICEVERE LA RIVISTA  
QLUB — NOTIZIARIO DEL QL.



IN COLLABORAZIONE CON LA:

DISTRIBUTTRICE PER L'ITALIA DEI PRODOTTI

**sinclair**



parametri attuali resta invariato rispetto a quello della chiamata (fig. 4).

## Parametri VARIABLE

Sono il mezzo per lo scambio di informazioni fra i due moduli, il chiamante e il chiamato. In questo caso i parametri attuali vengono modificati e il loro nuovo valore viene ritornato al programma, ai termini della procedura (fig. 5).

## Header e chiamata di funzione

L'header di una funzione contiene il nome della funzione, una lista di parametri che sono uguali a quelli visti per le procedure e il tipo del risultato (fig. 6).

Poiché la chiamata di una funzione ha come effetto la produzione di un valore risultato, è possibile richiamare le funzioni solo all'interno di espressioni.

Per assegnare alla funzione il suo risultato, ci deve essere, all'interno del block contenente il corpo della funzione, almeno un assegnamento con il nome della funzione sulla sinistra; in caso contrario il risultato di detta funzione sarebbe indefinito (fig. 7).

Vediamo ora un esempio di procedura senza parametri, che viene utilizzata dal programma principale per calcolare la somma di un dato numero con tutti i suoi predecessori:

```
PROGRAM
esempio;
  VAR a,b :
    INTEGER;
  PROCEDURE somma;
    VAR c :
      INTEGER;
  BEGIN
    (della procedura)
    a := 0;
    FOR c := 1 TO b DO
      a := a + c
```

```
END;
(della procedura)
BEGIN
  (del programma
  principale)
  b := 10;
  SOMMA
  (chiamata
  della procedura)
END.
(del programma
principale)
```

Vediamo anche un programma analogo che utilizza però una funzione:

```
PROGRAM ESEMPIO;
  VAR a,b,risultato :
    INTEGER
  FUNCTION SOMMA :
    INTEGER;
    VAR c :
      INTEGER;
  BEGIN
    (della funzione)
    a := 0;
    FOR c := 1 TO b DO
      a := a + c;
    SOMMA := a
  END;
  (della funzione)
BEGIN
  (del programma
  principale)
  a := 10;
  risultato := SOMMA
END.
  (del programma
  principale)
```

Vedremo in seguito alcuni esempi di procedure e funzioni più complesse nel contesto di programmi Pascal completi.

(segue da pag. 8)

Il tutto con il vantaggio di trasportare un programma in C da una macchina all'altra con modifiche minimali.

## Impararlo o no?

Tutto ciò naturalmente non significa che il C sia un paradiso del programmatore: essendo un linguaggio a livello medio-basso, non possiede certe funzioni sofisticate del basic, e chi è abituato a questo può trovarsi inizialmente a disagio; per esempio, manca un'operazione per confrontare direttamente stringhe di caratteri, e occorre scrivere un'apposita funzione. D'altro canto, ciò consente di avere un compilatore estremamente compatto.

Né vogliamo a tutti i costi convincervi della necessità di imparare il C: vivrete benissimo senza. State però certi che, se volete diventare programmatore professionista, nei prossimi anni (fino all'arrivo di un nuovo standard) farete inevitabilmente i conti con l'accoppiata C+UNIX.

Nel frattempo, anche con un umile Spectrum avete la possibilità di familiarizzare con il linguaggio dei grandi sistemi.

## Bibliografia

In italiano sul "C" sono reperibili per il momento due volumi: l'edizione italiana del manuale di Kernighan & Ritchie, gli autori del linguaggio, che è considerato la bibbia dell'argomento, contenente sia un'ampia guida all'uso, sia il manuale di riferimento per lo standard (Jackson 1985, p. 228, L. 21.000) e "Programmare in C", di C. Nowakowski (EPSI 1985, p. 136, L. 15.000), al quale però dobbiamo muovere un appunto prima ancora di vederlo (è uscito da pochissimo): la pubblicità esordisce infatti con "Il C è un linguaggio di programmazione ad alto livello..."; come abbiamo invece detto, "il C è un linguaggio relativamente a basso livello" (Kernighan, pag. 1 ed.it.). Questione di punti di vista?

Altri libri sul C, in inglese, sono reperibili presso le librerie specializzate delle grosse città.

Il manuale allegato al "C for the Spectrum" della Hi-Soft, peraltro ottimo, è anch'esso disponibile solo in inglese.







**VALHALLA  
LEGEND  
Spectrum 48K.**

Valhalla è un gioco di avventura veramente unico nel suo genere e tuttora insuperato nella grafica semovente, sebbene sia stato realizzato nel 1983 da un gruppo di 8 programmatori.

Per "grafica semovente" intendiamo il fatto che esistono nel mondo del Valhalla un mucchio di esseri che si muovono per i fatti loro, ciascuno dotato di propria personalità, e che possiamo realmente vedere "vivere" sullo schermo, in tempo reale. Come dice, a ragione, il manuale d'uso di 30 pagine, "Valhalla" può assicurare ore di divertimento, soltanto stando ad assistere agli avvenimenti che si svolgono parallelamente al-

le nostre azioni, nell'infinito numero di località visitabili.

Il nostro compito è di trovare 6 oggetti fatisi nascosti, individuabili solo in una certa sequenza, che possono essere anche in possesso di qualche personaggio mitologico, oppure celati in qualche scrigno o luogo oscuro.

Vi è un numero imprecisato di località visitabili, ciascuna con una grafica molto gradevole e suggestiva, anche se ovviamente molte località differiscono solo per qualche particolare tra di loro.

I 38 esseri del Valhalla, ciascuno col suo nome e con le sue caratteristiche di bontà/malvagità, abilità nella lotta, intelligenza e generosità, riportate nel manuale, sono divisi in «buoni» e «cattivi», e possiamo allearci ora con gli uni ora con gli altri. Sono: cani, draghi, uccelli, serpenti, semidei e semidee, dei.

Il programma dispone di un vocabolario con numero imprecisato di sostantivi e con 36 verbi/comandi tra i quali alcuni che ci permettono di comunicare, chiedere aiuto, trattare con gli altri personaggi.

Vi sono anche 21 tipi di oggetti disseminati, tra i quali armi di offesa e di difesa e cibo; quando impartisce l'ordine "eat food" il nostro omino realmente estrae il cibo e lo pone in bocca!

Il funzionamento del 23K di basic e 12K di assembly è una specie di multitasking; è possibile impartire comandi persino mentre un altro è ancora da eseguire o mentre sul video si sta lottando o facendo qualcosa.

Grafica: animata, scenografica, varia, coloratissima, ottima. 10

Comandi: buon set di comandi con editor di sintassi dei comandi. 9

Giocabilità: Un gioco sempre nuovo, spendibile su nastro. 10

```

9989 INPUT "SALVARE O PRINTARE (
S/P) ";s$: IF s$("<"s" AND s$("<"p
" AND s$("<"S" AND s$("<"P" THEN
GO TO 9989
9990 INPUT "DATI PRECEDENTI (S/N
) ";d$: IF d$("<"s" AND d$("<"n" A
ND d$("<"S" AND d$("<"N" THEN GO
TO 9989
9991 IF d$="s" OR d$="S" THEN P
OKE 58260,PEEK 58000: GO TO 9997
9992 INPUT "RIGA ";r: POKE 58000
,r
9993 INPUT "COLONNA ";c
9994 INPUT "LARGHEZZA ASSE ~X~ "
;l
9995 INPUT "ALTEZZA ASSE ~Y~ ";a
9996 POKE 58260,r: POKE 58261,c:
POKE 57773,l: POKE 57766,a
9997 IF s$="s" OR s$="S" THEN P
OKE 57782,64: GO TO 9999
9998 POKE 57782,74
9999 RANDOMIZE USR 57700
    
```

**Errata corrige**

*Nel commento al programma WINDOW pubblicato sullo scorso numero si faceva riferimento a un programmino basic di servizio, che doveva servire a esemplificare l'uso della routine. Per un incidente tecnico il listato di questo programmino è rimasto escluso. Lo pubblichiamo qui a fianco.*



"I dati numerici trattati dal calcolatore in virgola mobile sono sempre nel formato a 5 bytes, detto appunto *floating point*, ma nello Spectrum si distinguono due possibili rappresentazioni:

- *full floating point*, valida per qualsiasi valore intero o decimale, compreso tra 0.29387359E-38 e 1.701+38: il primo dei 5 bytes è l'esponente, gli altri 4 costituiscono la mantissa;

- forma semplificata, usata per numeri compresi tra -65536 e +65535, in cui il primo e il quinto byte sono sempre zero, il secondo è 0 per i numeri positivi e 255 per i negativi, il terzo e il quarto sono, nell'ordine, il byte meno e più significativo del numero reale, se è positivo, del suo complemento a 2, se è negativo.

Vi è poi una terza forma, detta *compressa*, utilizzata dal literal 52 (*stack-data*), che consente di rappresentare un valore in *fl. point* con un numero ridotto di bytes, da 2 a 5." (il passo è tratto dal libro "I se-

## Numeri in virgola mobile

di Giovanni Mellina

*greti dello Spectrum"*, opera dell'autore di questa utility, che potete richiedere alla nostra casa editrice).

Il breve programma qui presentato esegue la conversione di un valore numerico da uno dei seguenti formati agli altri due: decimale, full floating point, floating point compresso.

La conversione viene effettuata da una breve routine in linguaggio

macchina, sistemata nella linea REM all'inizio del listato.

La prima linea legge una variabile di sistema che individua la locazione a cui inizia la REM che segue: rende perciò il programma utilizzabile senza modifiche anche con l'interfaccia 1.

Digitate la REM con 20 caratteri qualsiasi: il codice macchina viene ricaricato a ogni RUN.

Un programmino di utilità estremamente versatile: esegue lo scroll orizzontale di una stringa di qualsiasi lunghezza, facendola scorrere da destra a sinistra sullo schermo dopo averla ingrandita, e ripetendola all'infinito (fino a che non viene dato un comando di break). Come una titolatrice, dite? Esatto.

La prima linea serve per individuare la locazione di inizio della REM che segue la linea stessa, e cui è contenuta la routine in linguaggio macchina, lunga 132 bytes: tanti sono i caratteri (qualsiasi) da inserire nella REM al momento della digitazione del programma.

La variabile NXLIN (23637/8) contiene infatti l'indirizzo della prossima linea di programma - prossima a quella puntata al momento dal cursore di programma: questo consente, avendo una routine totalmente rilocabile come questa, di in-

## Scroll orizzontale

di Mauro Piazzola

serire in qualsiasi punto del listato, ed evita che sorgano problemi quando interviene la ROM ombra dell'Interfaccia 1.

Battete il programma facendo attenzione a trascrivere esattamente tutti i DATA, che verranno automaticamente caricati nella REM, non appena date RUN.

Quindi vi viene chiesta la stringa da far scorrere. Con R si ripete il

RUN, per un'altra scritta, con E si esce.

La routine non tocca gli attributi correnti, perciò potete predisporre qualsiasi combinazione di INK e PAPER, ottenendo effetti spettacolari. Il movimento da sinistra a destra è una "scivolata" molto fluida e gradevole, completamente priva di scatti. Ottimo per pubblicità in vetrine.



## Chiamate al sistema operativo

Con questo numero iniziamo una serie di articoli dedicata all'analisi del sistema operativo del QL, il QDOS, illustrando le più importanti routine.

Per ciascuna di esse riporteremo i parametri di chiamata, i parametri di ritorno, una dettagliata spiegazione del funzionamento e i codici necessari per accedervi. Prima di questo, è però opportuno tracciare a grandi linee lo schema del sistema operativo del QL e il modo di accedervi da linguaggio macchina.

Innanzitutto la domanda di rito: cos'è un sistema operativo? Essenzialmente si può affermare che un sistema operativo è quell'insieme di routines d'utilità che si occupano di tutti i compiti elementari di una macchina, come scrivere un carattere sul video o leggere la tastiera, e che amministra l'uso della memoria e delle periferiche collegate al computer.

Nel QL il sistema operativo si chiama QDOS e la caratteristica che meglio lo distingue nel gran numero di sistemi operativi delle macchine della sua stessa fascia di prezzo è la capacità di supportare il *multitasking*, ovvero la facoltà di far girare quasi simultaneamente più di un programma in ling. macchina; così avviene, in configurazione base, con il sistema operativo e il basic.

Questa caratteristica nasconde delle possibilità veramente notevoli,

per esempio negli adventures, ma anche in moltissime altre applicazioni, e bisogna notare che finora nessuna macchina al di sotto dei 3 milioni ne dispone, a riprova del fatto che il *multitasking* richiede una notevole potenza di base per poter essere implementato, e sul QL, grazie al processore a 32 bit e ai 128K di memoria, tutto ciò diventa possibile a un prezzo estremamente basso.

Il QDOS è diviso in varie sezioni, ognuna dedicata allo svolgimento di un compito ben preciso nell'ambito della macchina: gestione del *multitasking* (SCHEDULER), controllo della memoria, controllo delle risorse, uso delle risorse, routines d'interrupt. L'accesso a queste sezioni è possibile mediante una lunga serie di routines di gestione costituenti il cosiddetto *Kernel* del sistema operativo.

Ci sono due differenti modi per chiamare una routine, a seconda che essa debba funzionare in modo USER o in modo SUPERVISOR: nel primo caso si usano dei VETTORI (puntatori a 16 bit che indicano la locazione d'inizio della routine in questione), nel secondo della TRAP.

Per esempio, per chiamare il vettore situato alla locazione 110H, useremo queste istruzioni:

```
...
(blocco d'istruzioni che caricano i
parametri richiesti nei giusti registri)
```

```
...
MOVE.W $110,A2
JSR (A2)
```

ovviamente il registro A2 può essere sostituito con un altro a piacere fra quelli che non devono essere passati come parametri alla routine.

Una TRAP si chiama invece in modo più semplice: ad esempio per chiamare la routine MT.INF, il cui codice è (d0=0, trap 1), si userà:

```
...
(blocco d'istruzioni che caricano i
parametri richiesti nei giusti registri).
```

```
...
MOVEQ $0,D0
TRAP 1
```

Con il 68008 ci sono 16 TRAPS, da 0 a 15, a disposizione del sistema. Mentre le TRAPS da 0 a 4 compare sono riservate al sistema operativo, le altre sono a disposizione dell'utente, che può riprogrammarle mediante un'altra routine del QDOS.

Il procedimento generico per chiamare una TRAP è quindi il seguente: dapprima si caricano i giusti registri con i parametri da passare alla routine; poi si seleziona la routine nell'ambito della TRAP inserendone il codice in D0; infine si esegue la TRAP.

I parametri di ritorno, per entrambi i sistemi, sono contenuti nei registri del processore o, in certi casi, nello stack. Il codice d'errore di trova invece in D0, e il valore riportato è quello standard QDOS (abbiamo pubblicato un articolo sull'argomento).

Se D0=0 al ritorno della routine, è tutto OK; altrimenti c'è qualche errore a cui la routine deve reagire analizzandone il codice. Non è necessario eseguire dei controlli su D0, in quanto questi vengono già eseguiti dal QDOS, che prima di passare il controllo al programma chiamante esegue un TST.B d0; di conseguenza i flags vengono influenzati dal valore di d0.

In pratica, per accertarsi che non c'è stato alcun errore nello svolgimento della routine, il programma chiamante deve semplicemente operare così:

```
...
(chiamata della routine)
BNE label
```

```
...
...
```

(istruzioni da eseguire nel caso non ci siano stati errori)

```
...
label
```

(istruzioni da eseguire in caso di errore).

Nei prossimi numeri inizieremo a illustrare le routines più importanti del QDOS.

SPAZIO QL

# Hard-copy del video

Bolla, Mussini & Zonin

La riproduzione su carta di una schermata (l'*hard-copy*, appunto) è una delle opzioni che la maggior parte degli utenti considera, a ragione, irrinunciabile; non si tratta solo di poter stampare la schermata di un gioco - sarebbe poco più che uno sfizio da hobbista: poter trasferire sul foglio grafici e tabelle è estremamente utile in qualsiasi lavoro. E vedremo tra poco che proprio il pacchetto grafico-gestionale EASEL consente di realizzare copie dello schermo.

Anche i più masochisti dovrebbero già essersi resi conto che tentare soluzioni in basic richiede la pazienza di Giobbe: 32 kilobytes da trasferire, quando i 7k scarsi dello Spectrum impegnavano la stampante per oltre 10 minuti (parliamo di basic su stampanti non dedicate), dissuaderebbero chiunque.

È quindi necessario lavorare in linguaggio macchina.

Fortunatamente, EASEL contiene la routine che ci serve, e il poco lavoro da fare consiste nell'isolamento di tale software dal resto del pacchetto, per poterlo utilizzare in tutte le situazioni.

Il tempo di esecuzione di una copia è di circa 55 secondi (dipende anche dalla stampante usata): un tempo decisamente eccezionale, se pensate che si tratta di un'*hard-copy* "intelligente", cioè non sem-

plimente bianco e nero, ma con una scala di grigi, secondo i colori, ottenuta con diverse densità di punti.

La routine è gestita con la tecnica detta di *overlay*: il termine sta a indicare che una stessa area di memoria del computer viene utilizzata per più routines, che risiedono tutte sul nastro (o sul disco), e vengono caricate di volta in volta al momento dell'utilizzo.

Questo, come potete intuire, facilita molto anche il lavoro di individuazione e isolamento della routine stessa:

- a - ponete EASEL nel drive 2
- b - ponete una cartuccia con un po' di nastro libero (basta un paio di settori) nel drive 1
- c - eseguite uno dei seguenti comandi (uno solo), secondo la stampante posseduta:

per Epson FX-80 e compatibili  
COPY MDV2 GPRINT PRT TO MDV1 HARDCOPY (enter)  
oppure, per Integrex 132 & c.  
COPY MDV2 INTGX132 PRT TO...

oppure, per Okimate 80 & c.  
COPY MDV2 JX80 PRT TO...  
Nell'eventualità di un messaggio *Bad or changed medium*, la colpa è, nella quasi totalità dei casi, della cartuccia destinataria: sostituirla.

Per caricare in memoria la routine, occorre eseguire preventivamente

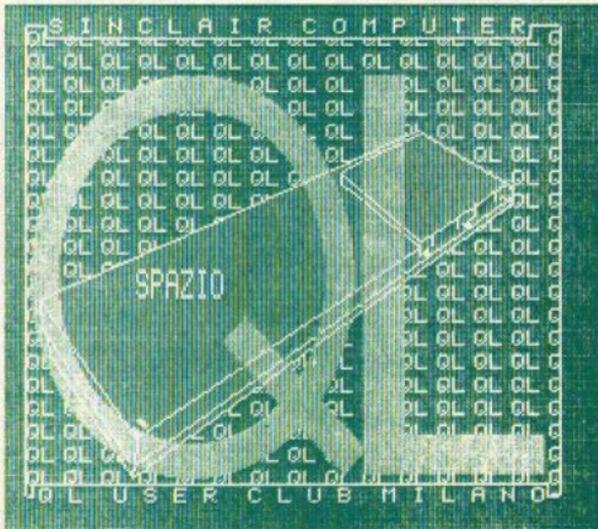
HCOPI = RESPR (1000)  
con cui il QL riserva alla routine un'area di RAM, quindi, di seguito  
LBYTES MDV1 HARDCOPY, HCOPI

Quando avrete la schermata pronta da stampare, date semplicemente

CALL HCOPI  
Inutile dire che la variabile HCOPI non deve essere alterata.

Potete anche scrivere una breve PROCedura che effettui la chiamata, riducendo ancora il numero di tasti da premere.

Come ultima cosa ricordiamo che la velocità di trasmissione sulla porta SER1 è di 9600 baud: volendo alterarla basta usare l'apposito comando basic BAUD n.



SPAZIO QL

# Super basic

di Marco Mussini

Prima di cominciare a parlare dei devices SCR e CON, chiariamo un particolare riguardo al filename da usarsi quando si apre un canale.

Con SCR e CON ci sono rispettivamente quattro e cinque parametri da specificare in coda al device name (per es. SCR\_100 x 200a20 x 30).

Tuttavia il QDOS ha un default per ciascuno di questi parametri (vedere più avanti) e permette quindi di saltarli tranquillamente tutti o solo in parte; quando vengono ignorati in blocco, è tutto chiaro:

OPEN #3, SCR

apre una finestra di dimensioni e posizioni standard collegandola al canale 3; ma quando se ne specifica solo qualcuno il discorso si complica. In questo caso, il QDOS controlla quali parametri mancano e quali ci sono, dopodiché ricostruisce il filename completo usando i dati dell'utente (dove ci sono) o il relativo valore di default (dove mancano). Per esempio, supponiamo di impostare

OPEN #4, SCR\_a20

il QL fonderà questo filename con il default fino a completarlo. Seguiamo passo passo il ragionamento del QDOS (sopra il filename, sotto il default)

SCR\_a20

SCR\_448 x 180A32 x 16

SCR\_448 x A20

SCR\_448 x 180A32 x 16

SCR\_448 x 180A20

SCR\_448 x 180A32 x 16

SCR\_448 x 180A20 x 16

SCR\_448 x 180A32 x 16

Come si vede, a un certo punto il QL capisce che vogliamo sostituire il valore di default della coord. orizzontale (32) con uno deciso da noi (20), quindi elimina il rispettivo valore di default prendendo per buono il nostro; alla fine ottiene il filename completo, che altro non è se non il risultato di una fusione fra il default e i dati che abbiamo specificato.

## DEVICES DISPONIBILI SUL QL INESPANSO

### IL DEVICE SCR

#### A) GUIDA DI RIFERIMENTO

Tipo di dati: 8 bit

Direzione dati: solo output, non bufferizzati.

Caratteri particolari: chr\$(10) è A CAPO (LF, Line Feed), i restanti caratteri non compresi fra il codice 32 e il 191 vengono visualizzati con una griglia per segnalare che il carattere non è definito da alcun pattern.

Sintassi del filename per OPEN:

SCR\_larghezza x altezza A posizione x X posizione y

Esempi validi:

SCR\_100 x 100a36 x 22

SCR\_100 x 100A36 x 22

scr\_30 x 40a50 x 60

scr A2 x 2

scr

Esempi non validi:

SCR100 x 100a20 x 50 (manca il separatore)

SCR\_20 x 30a20 x 30 (sbagliato il device name)

SCR\_10a30 x 14a22 (errate le lettere di separazione)

SCR\_\_2 x 2a2 x 2 (troppi separatori)

SCR\_10 x 10a10 x 10 x 20 (troppi parametri)

Default:

SCR\_448 x 180a32 x 16

Tipo di device: output su schermo (in modo testo: la grafica è un discorso a parte, che tratteremo in una prossima puntata)

#### B) CARATTERISTICHE

SCR è un device di solo output: inutile quindi cercare di effettuare degli INPUT rivolti ad esso.

I parametri specificati in coda al device name determinano la dimensione e la posizione della finestra

cul il canale andrà a collegarsi; l'unità di misura è il pixel. Le finestre non possono essere più grandi di tutto lo schermo, e non devono "sbucarne fuori", nemmeno in parte. Le coordinate dello schermo, in entrambe le risoluzioni, sono 0-511 in orizzontale e 0-255 in verticale, con l'origine (0,0) situata in alto a sinistra. Sul TV l'area visibile è generalmente quella contenuta in una finestra larga 448 e alta 240 pixels, situata 32 pixel a destra del margine sinistro e 16 sotto il margine superiore. Con il TV si perdono infatti 32 pixels a sinistra, altri 32 a destra e 16 in alto. Nessuna differenza nella fascia bassa, fatta eccezione per la minor larghezza.

I caratteri inviati a una finestra vengono visualizzati in sequenza da sinistra verso destra; le eccezioni sono due:

1) I caratteri hanno raggiunto il limite destro;

2) il device driver riceve un carattere ASCII 10 (Enter).

In questi due casi il computer effettua un CR e un LF, ovvero va a capo. A differenza dello Spectrum, se lo schermo è pieno lo scroll viene eseguito senza chiedere conferma all'operatore, tuttavia, se ve n'è il bisogno, si può bloccare il computer premendo CTRL-F5, che blocca non solo lo scroll, ma qualsiasi cosa giri sotto QDOS (con qualche eccezione, anche il linguaggio macchina).

In modo 8, i pixel sono la metà (256 per riga), eppure le coordinate vanno ugualmente da 0 a 511. La cosa si spiega se si considera che i pixel sono contati due volte: così, se in modo 4 i punti (100, 100) e (101, 100) sono distinti, in modo 8 sono soltanto due modi di indirizzare lo stesso pixel. Ciò permette una perfetta compatibilità dei comandi grafici in modo 8 e in modo 4: se così non fosse, i programmi, se fatti girare in modo 4, dovrebbero raddoppiare le coordinate orizzontali.

Inoltre ne discende che, aprendo una finestra con le stesse coordinate e dimensioni in modo 4, si ha esattamente lo stesso effetto che si sarebbe ottenuto in modo 8.

Lo schermo è una periferica piuttosto complessa: il QDOS riserva ad ogni finestra aperta uno spazio supplementare al consueto channel definition block (window definition block), che contiene informazioni sui colori, sulla forma e dimensione dei caratteri, sullo stato di OVER, sul BORDER, sulla grafica, sugli

## SPAZIO QL

stipples, sulle dimensioni della finestra e così via. Poiché ogni finestra ha un suo *window definition block* personale, le finestre sono completamente indipendenti le une dalle altre, e possono addirittura avere set di caratteri diversi, senza interferire. In questo stesso numero riportiamo due tabelle sull'organizzazione del *channel definition block* e *window definition block*, tratte dall'"Advanced User Guide" di A. Dickens, edita da Adder, Cambridge.

Per chi volesse compiere esperimenti sull'argomento, ricordiamo inoltre che il *channel definition block* del canale 0 inizia alla locazione 167424 e che quelli dei canali 1 e 2 cominciano rispettivamente 256 e 512 bytes più avanti.

Non si può mai sapere con sicurezza, invece, la dislocazione dei *channel definition blocks* relativi a canali aperti dopo i primi tre, perché il QDOS li mette dove gli fa più comodo e non dove la logica suggerirebbe di cercarli.

### IL DEVICE CON

#### A) GUIDA DI RIFERIMENTO

**Tipo di dati:** 8 bit

**Direzione dati:** input/output. L'input è bufferizzato.

**Caratteri particolari:** gli stessi di SCR; inoltre, premendo CTRL-C, si passa da una finestra all'altra nel caso che più programmi stiano girando in multitasking, aspettando dati ognuno nella sua window.

**Sintassi del filename per OPEN:**

CON \_ larghXaltXapoxXpoxylunghezzabuffer

**Esempi validi:** vale quanto già detto per SCR. Si ricordi solo di non trascurare il separatore fra l'ultimo parametro della finestra e quello di estensione buffer.

**Esempi non validi:** vedi sopra; attenzione solo a questo caso:

CON \_ \_128

che è perfettamente valido e crea una finestra di dimensioni default con un buffer di 128 bytes.

Infatti il QL, incontrando due separatori, capisce che ciò che troverà dopo il secondo separatore è il parametro sull'estensione del buffer.

Con SCR, dove è invece previsto

un solo separatore in tutto il filename, trovandone due QL visualizza un messaggio di errore.

**Default:**

CON \_ 448 x 180a32 x 16 \_ 128

**Tipo di device:** svolge in pratica le funzioni di un terminale: riceve dati da una tastiera, visualizzandoli sul monitor (la finestra) man mano, e inserendoli in memoria (il buffer), aspetta la pressione di ENTER, li trasmette al computer (input) e visualizza e eventuali sue risposte (output).

Per non tenere impegnato il computer, i dati vengono spediti tutti insieme (si dicono quindi BUFFERIZZATI) dopo essere stati raccolti, appunto, in buffer la cui dimensione è determinabile dall'utente. Il default è di 128 bytes, ma si possono specificare valori ben maggiori.

È questo buffer che permette al QL di prendere ordini mentre è occupato con qualche altra attività per poi eseguirli appena libero. Il buffer può anche riempirsi senza

aver ricevuto nessun ENTER; in questo caso, il QDOS lo svuota e visualizza il messaggio d'errore "buffer full".

#### B) PARTICOLARITÀ

Resta solo da aggiungere che il device CON supporta un flusso bidirezionale di dati e che quindi permette l'uso di INPUT. Si possono effettuare INPUT dai canali 0, 1 e 2, proprio perché non sono dei semplici SCR ma del CON; provate a digitare

```
O P E N # 1 ,
SCR _ 448 x 200A32 x 16
PRINT "PROVA OUTPUT"
INPUT "PROVA INPUT"; AS
```

e vi accorgete che il QL s'impunta quando gli si chiede di ricevere dati da un SCR, che è unidirezionale. Per ristabilire il consueto ordine di cose date

```
OPEN # 1, CON _ 448 x 200a3-
2 x 16 _ 128.
```

Per verificare che tutto è a posto provate a comandare un INPUT as.

## JS, il QL si evolve

Dopo le versioni PB, FB, AH, JM (le ultime due, apparentemente equivalenti, costituiscono la stragrande maggioranza del parco macchine installato in Italia), la Sinclair ha approntato una nuova versione di basic e sistema operativo, che equipaggia i nuovi QL venduti in Inghilterra a partire da settembre.

La ROM JS, così si chiama l'ultima release, presenta sostanziali migliorie rispetto alle precedenti: per il basic abbiamo la scomparsa di quasi tutti i famigerati bugs rimasti anche nella JM e l'implementazione, sembra, di buona parte dell'ottimo "QL-Toolkit" di Tony Tebby & QJUMP; le novità più significative sarebbero quindi l'accesso random ai files su Microdrive, a gestione, eliminazione, attivazione, analisi dei Jobs, e quindi del multitasking, direttamente da Basic; inoltre nuove varianti di EXEC adat-

te a concatenare i jobs e a permettere loro di scambiarsi dati, conversioni binario/decimale - decimale e viceversa, funzioni per incolonnare correttamente i numeri (una specie di PRINT USING) e così via.

Ma soprattutto è stata finalmente implementata la gestione degli errori, resa possibile da alcuni nuovi comandi; si tratta di un sistema molto flessibile e potente: in pratica, ogni volta che incontra un errore, il QL, anziché fermarsi, esegue il gruppo di linee compreso tra WHEN ERROR e END WHEN; questa routine può regolarsi in base al tipo d'errore capitano, poiché il basic JS mette a disposizione due appositi funzioni, ERLIN ed ERNUM, che ritornano rispettivamente il codice e il numero di linea dell'ultimo errore capitato.

Quando al QDOS, le stesse voci parlano di incrementi delle prestazioni dei Microdrives, ovviamente in termini di velocità, e dell'eliminazione dei pochissimi bugs rimasto.

Sarebbero infine state implementate alcune utilities prima diftose o mancanti.

In Italia la ROM JS non è ancora disponibile.



In edicola  
questo mese

**SINCLAIR**  
COMPUTER

PRESENTA

# Tutti i segreti dello SPECTRUM

4 passi nella ROM: come utilizzare  
le routine più importanti  
del sistema operativo  
a cura di Giovanni Mellina



 I LIBRI DI  
systems

Vivi anche tu l'avventura del

**CAMEL  
TROPHY**

fantastico "ARCADE"

per ZX - SPECTRUM 48 K



SUPPLEMENTO  
al N. 12<sup>o</sup>  
di RUN

**OGGI IN TUTTE LE EDICOLE**

**CAMEL**  
PRODUCTIONS



IN  
COLLABORAZIONE  
CON

**AQUARIUS EDIZIONI**

Nel caso non lo trovaste nella Vostra edicola, potrete riceverlo inviando vaglia postale di Lire 12.000 ad Aquarius Edizioni  
via Leopardi n° 9, 20123 Milano, specificando il Vostro nome, cognome e indirizzo.

