

L'illusione di aver messo gli occhi sulle locazioni di memoria dura però assai poco, in quanto è immediato scoprire che si tratta di un banale controllo di parità, utile solo per pescare al volo gli errori durante la lettura da parte del PC-1211.

In realtà di locazioni di memoria non si trova proprio traccia in nessun angolo del file: pertanto ogni programma viene caricato risalendo a ritroso i registri della memoria flessibile fino a dove è necessario: addio speranza di rilocazioni! Lo stesso avviene per le variabili, al punto che se uno per esempio salva i dati con

```
PRINT # "FILE";A(100)
```

si deve ricordare il numero 100 se desidera ritrovarsi al loro posto, perché come è noto, l'istruzione

```
INPUT # "FILE"
```

li sistema a partire dalla variabile A(1) in poi.

Organizzazione del titolo dei file

Ad orecchio potrebbe sembrare che la struttura dei file di programma sia diversa da quella dei file di dati; si sente comunque che entrambi sono suddivisi in record di lunghezza variabile a seconda del tipo e che la pausa che li separa ha circa la loro stessa durata.

Non a tutti sarà sfuggito inoltre che il primo record è praticamente uguale nei due casi: in effetti si tratta di un pacchetto di 24 byte che inizia con un gruppo di 4 (in funzione di identificatore), atto a segnalare

se il seguito sarà un programma (da allocare a partire dal più alto degli indirizzi dell'area della memoria flessibile) o dei dati (da allocare salvo diverse prescrizioni, comunque non residenti nel file, a partire dal più basso degli indirizzi della memoria fissa); in figura 4 si forniscono i codici identificatori rilevati.

CSAVE	"FILE"	FF80
PRINT #	"FILE"	0F8F
PRINT #	"FILE"; A(I)	F08F

Figura 4 - Identificatori del tipo di file del PC-1211

Successivamente si trovano sette coppie di byte che contengono il nome del file: se il nome è più breve di sette caratteri, ci sono comunque 14 byte e quelli non utilizzati codificano il carattere corrispondente al tasto SPC.

Infine vengono i famosi due byte di controllo di cui si è parlato prima e su cui ritorneremo; essi operano il loro controllo solamente sui 14 byte del titolo e non sui primi quattro.

A questo punto interviene la prima pausa durante la quale il segnale mantiene la frequenza costantemente a 4 kHz ed il pin 9 resta al livello logico Low.

Dopo ciò inizia il primo record vero e proprio; la sua lunghezza può avere solo due valori:

18 byte per il file delle variabili;

180 byte per il file dei programmi (salvo l'ultimo che può essere più corto).

Quanti sono questi record? Per i dati il conto è presto fatto: supponiamo che la risposta alla istruzione MEM sia 1264 STEP 158 MEMORIES e che il file debba essere generato con la generica istruzione

```
PRINT # "FILE"; A(I)
```

Se I=1 l'istruzione si semplifica, come è noto in

```
PRINT # "FILE"
```

Definiamo le due variabili:

$$S = 1264$$

$$M = 158$$

allora il numero D dei record del file dei dati sarà:

$$D = 1 + M + 26 - (I - 1) = 185 \text{ record}$$

infatti, oltre al primo record del titolo bisogna ricordare che ci sono anche le 26 variabili della memoria fissa.

Il calcolo del numero P dei record del file dei programmi è un pochino più laborioso: definiamo altre due variabili e cioè T (numero totale di step disponibili) e X (variabile di servizio).

$$T = 1424 \text{ step max}$$

$$X = (T-S)/80 = 1.95$$

il numero dei record desiderato sarà (in Basic PC-1211):

$$P = 1 + \text{INT } X + ((X - \text{INT } X) <> 0) = 3 \text{ record}$$

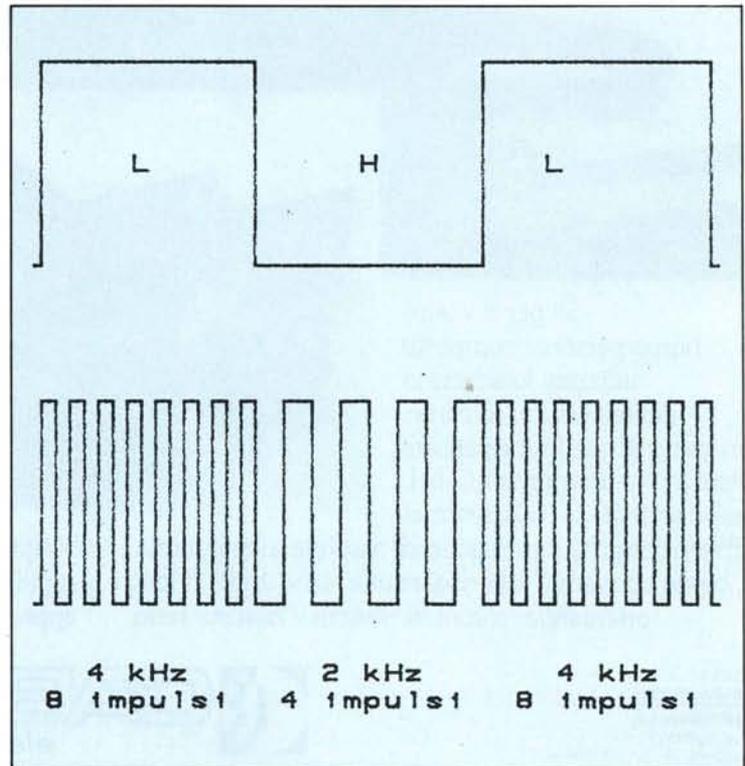
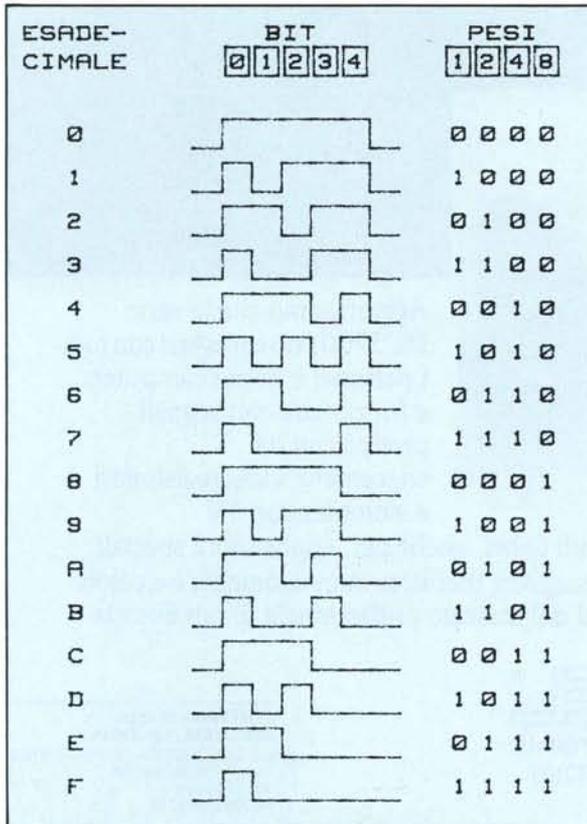
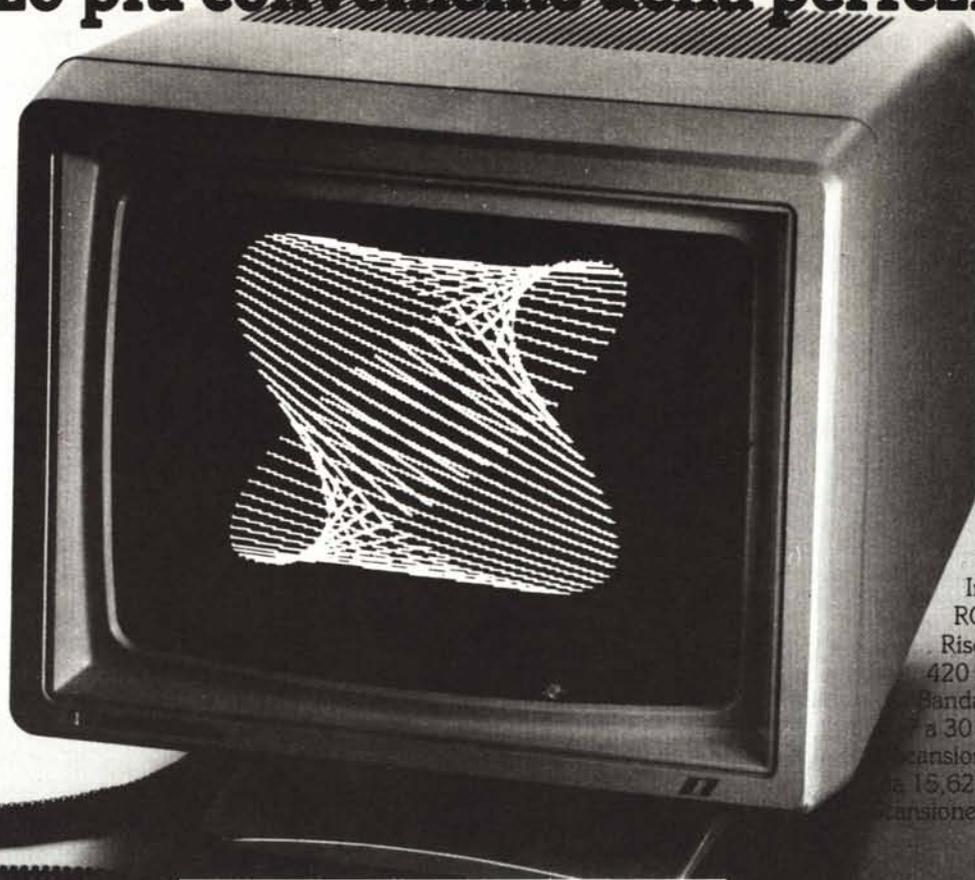


Figura 3 - Il codice bitonale del PC-1211.

Figura 2 - Il codice binario del PC-1211.

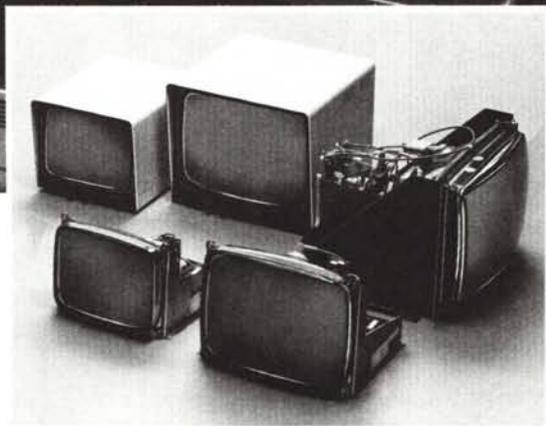
Monitor Cabel.

Il prezzo più conveniente della perfezione.



MC 3700

Ingressi: PAL/C-64;
RGB; PAL/RGB.
Risoluzione da:
420 a 800 PIXEL.
Banda passante da:
a 30 MHz.
Espansione orizz.
a 15,625 a 32 KHz.
Espansione vertic. 50/60 Hz.



Se per il vostro
home-personal computer
utilizzate lo schermo
del televisore, riflettete.

Con meno di quello che pensate
potete avere un monitor Cabel.

La nuova serie MC 3700 unisce al
raffinato design caratteristiche di assoluta avanguardia:
basso consumo, alta risoluzione, affidabilità, video
orientabile, comandi frontali e non sul retro.

Aggiungiamo che la serie
MC 3700 può collegarsi con tutti
i personal e home computers
e funzionare con segnali
provenienti da
telecamere, videoregistratori
e sintonizzatori TV.

Scegliere un Cabel, anche per applicazioni speciali,
significa scegliere monitors monocromatici e a colori
apprezzati dal mercato professionale di tutt'Europa.

CONCESSIONARI
ED ASSISTENZA TECNICA

MILANO E PROVINCIA

• BRESCIANI AMEDEO
Via A. Stoppani, 34 - 20128 Milano
Tel. 02/2043459

• TECHNEX s.r.l.
Via Teocrito, 46 - 20128 Milano
Tel. 02/2575315

• EMILIA ROMAGNA - MARCHE
• ONDAELLE s.n.c.
Via Faccini, 4 - 40128 Bologna
Tel. 051/373513 - 359649

• LIGURIA
• R. e R. ELECTRONICS s.r.l.
Via F.lli Canepa, 94
16010 Serra Riccò - GE
Tel. 010/750729 - 750866
Telex 216530 COGE I

• TOSCANA - UMBRIA
• FGM ELETTRONICA s.r.l.
Via Silvio Pellico, 9/11
50121 Firenze
Tel. 055/245371
Telex 573332 FGM I

• LAZIO
• HI-REL s.r.l.
Via Amatrice, 15
00199 Roma
Tel. 06/8395671 - 8395581
Telex 614676

• GIU. PA. R.
di G. Pastorelli e figli
Via dei Conciatori, 36
00154 Roma
Tel. 06/5758734

CAMPANIA - PUGLIA - BASILICATA - CALABRIA

• C.F. ELETTRONICA PROFESSIONALE
Corso V. Emanuele, 54
80122 Napoli
Tel. 081/683728

SICILIA

• RICCOBONO EMANUELE
Via Onorato, 46
90139 Palermo
Tel. 091/331464 - 325813

 **CABEL**[®]
electronic

24035 CURNO (Bergamo) - Tel. 035/612103
Telex 316370 CABEL I



software MBASIC

Come naturale continuazione della scorsa puntata, in questo numero estenderemo le nostre considerazioni riguardanti la gestione dei file per poter essere in grado, alla fine, di scegliere quale delle due strutture (file sequenziali o file random) risolve meglio il nostro problema.

I File random

Questi tipi di file prendono il nome dalla modalità con cui si accede ai singoli record: in questo caso tale accesso è "casuale" (random) nel senso che per leggere un certo record non siamo costretti a leggerne altri: abbiamo infatti visto la scorsa volta che, nei file sequenziali, l'accesso ad un dato non può avvenire se non si è effettuato l'accesso a tutti i dati "precedenti". Nel caso dei "random" si possono leggere qua e là i vari record del file senza limitazioni di sorta, ed inoltre scompare la distinzione tra accesso in lettura e accesso in scrittura: si parla infatti di accesso generico ad un dato per effettuarne indifferentemente la lettura e l'aggiornamento, avendosi in ogni caso, come vedremo, un'unica apertura fisica del file.

Proseguendo nell'analisi di questo tipo di file vediamo un'altra fondamentale differenza rispetto ai sequenziali: la lunghezza del record.

Nel nostro caso, tutti i record *devono* avere sempre la stessa lunghezza il che comporta che bisogna calcolare a priori e "una tantum" la lunghezza del record.

Ora se un certo record risultasse più corto, apposite istruzioni consentono di inserire degli spazi bianchi all'interno del record stesso, in modo da "estenderlo" al punto giusto.

Questo fatto già comporta l'inconve-

niente notevole dato dallo spazio inutilizzato.

Se invece abbiamo a che fare con un record che inaspettatamente risulta più grande del record base, allora dovremo effettuare delle scelte riguardo a quale parte dell'informazione contenuta nel record stesso deve essere forzatamente troncata.

Comunque già si vede bene che la prima difficoltà nella gestione dei file random è la determinazione della lunghezza effettiva del record.

Basti pensare che in molti casi in cui non si ha una precisa idea sulla grandezza dei record si sarà portati (e costretti) a sovrastimare l'ampiezza del record, riducendo così il numero di tali record registrabili su un dischetto. Parlando della lunghezza del record, riguardo al valore massimo in assoluto si può dire che esso può essere modificato all'atto della chiamata dell'MBASICS stesso, secondo le metodologie già descritte nella rubrica "I trucchi del CP/M" nei nn. 23 e 24 di MCmicrocomputer.

Ad esempio, se da ambiente CP/M (contraddistinto dal ben noto "A >") digitiamo

MBASIC/S:300

stabiliamo una volta per tutte che il valore massimo della lunghezza dei record è posto a 300, poi quando andremo a gestire il nostro file random, con un'apposita istruzione, potremo ancora diminuire tale valore, ma mai aumentarlo!

A questo proposito l'esperienza maturata per lungo tempo su un sistema Osborne I ci consente di dare ai lettori un suggerimento, sulla base di alcune considerazioni.

Nel nostro caso si doveva gestire un certo tipo di file random dove per comodità era stata posta l'ampiezza massima del record a 255 byte.

Ora, fatti i dovuti conti, la lunghezza effettiva del record era invece risultata di 150 byte. Ciò ha comportato che il sistema operativo ed in particolare la versione 5,21 del MBASIC non eliminava quei 105 byte rimanenti, utilizzandoli per altri record, ma viceversa dal momento che nell'Osborne I i settori logici sono di 256 byte, lasciava della "garbage" in tali byte inutilizzati. Ora tale "garbage" conteneva tra l'altro un bel po' di "carriage return" che, come nel caso dei file sequenziali, separano i vari record: questo fatto ha comportato enormi problemi quando lo stesso file è stato in seguito gestito con il programma dBase II.

La morale insegna dunque che per non avere problemi conviene stabilire una certa lunghezza del record ed invocare l'MBASICS con l'opzione "S" per poi riconfermare, all'atto dell'apertura del file, tale valore.

La gestione dei file random

Facendo riferimento alla tabella (analogamente a quella già vista per i file sequenziali), vediamo quali sono le operazioni da compiere. Innanzitutto ora la tabella è unica e si riferisce contemporaneamente alla scrittura dei record. Come vedremo tra breve, infatti, con i file random potremo, con lo stesso programma, aggiungere, leggere e scrivere record, il tutto con un'unica apertura ed un'unica definizione dello spazio costituente il record.

Ma andiamo con ordine: supponiamo, per semplicità, di riferirci al file dell'esempio di cui abbiamo parlato la scorsa puntata.

Supponiamo perciò di voler gestire una rubrica telefonica, questa volta con un file random.

Dobbiamo innanzitutto stabilire la lunghezza dei vari campi che costituiscono il record: tali campi saranno associati ad altrettante variabili stringa, di lunghezza da prefissare una volta per tutte e che nel nostro caso saranno C\$, N\$, I\$, T\$, rispettivamente per il cognome, il nome, l'indirizzo ed il numero di telefono.

Supponiamo di voler riservare 20 caratteri per il cognome, 20 per il nome e 30 per l'indirizzo.

Supponendo poi di avere numeri di telefono contenuti in variabili reali, la relativa stringa T\$ sarà formata da 4 byte e non già da 6 (tanti quante sono le cifre) come nel caso sequenziale.

Ciò si spiega facilmente considerando che una variabile reale viene codificata dall'MBASIC in 4 byte, e come tale viene gestita dai file random.

Inoltre se la variabile è intera avremo un'occupazione di 2 byte, mentre per una a doppia precisione avremo ben 8 byte: tutto questo, è evidente, indipendentemente dal "numero di cifre" che compongono il numero dato.

Ad esempio il valore 9507, formato da 4 cifre e perciò codificabile ASCII con 4 byte (caso dei file sequenziali) ora viene codificato con 2 byte esadecimali (25H e 23H) e come tale considerato come stringa.

Questa conversione si ottiene con le istruzioni MKIS, MKS\$ e MKD\$, rispet-

tivamente per variabili intere, in singola e in doppia precisione.

Ad esempio supponiamo che la variabile intera A% valga appunto 9507: per generare una stringa S\$ da immettere in un campo di un record useremo dunque l'istruzione MKIS:

S\$ = MKIS(A%)

In questo caso, dato che il valore di A% in memoria è posto come 23H e 25H (cioè prima il byte meno significativo e poi il più significativo), la MKIS costruisce una stringa lunga 2 byte di valore appunto 23H e 25H: se per curiosità andiamo a stampare il contenuto di S\$ troveremo "#%" e cioè i caratteri corrispondenti ai valori esadecimali 23He 25H e cioè 35 e 37 decimali. Tutto va come se avessimo costruito S\$ con CHR\$(35) + CHR\$(37).

Apertura e definizione dei campi

Dopo aver stabilito su quali variabili lavoreremo e soprattutto dopo averne definita la lunghezza in byte, siamo pronti ad aprire il nostro file, che supporremo chiamarsi "RUBRICA".

Tale operazione si effettua con una OPEN dove con "R" si indica la volontà di operare con file random, mentre per il numero del buffer e il "nomefile" valgono le stesse considerazioni dei file sequenziali: numero del buffer è un numero compreso

tra 1 e 15 e "nomefile" è nel nostro caso RUBRICA.

Una particolare attenzione riguarda il campo (opzionale) "lunghezza", che consente, come abbiamo visto precedentemente, di indicare la lunghezza effettiva del record: è facile verificare che tale campo non avrebbe alcun senso nel caso dei file sequenziali ed infatti non esiste nella loro istruzione "open". Fatto ciò possiamo (anzi dobbiamo!) specificare quali sono i campi del record e qual è la loro lunghezza: lo facciamo con l'istruzione FIELD nella quale compare anche il solito "numero buffer".

Nel nostro caso, avremo (vedasi anche il programma riportato) FIELD 1, 20 AS C\$, 20 AS N\$, 30 AS I\$, 4 AS T\$.

Ora bisogna fare grande attenzione sulle variabili usate per questa definizione: queste variabili vanno "gelosamente" riservate, nel seguito del programma, ai campi del record e non si possono usare per altri scopi.

Se invece ad esempio usiamo nel programma la variabile N\$ per generare generiche stringhe, quando poi useremo N\$ per indirizzare il 2° campo del record andremo a puntare la zona di memoria utilizzata per le stringhe invece della zona riservata al buffer: ciò comporterà evidenti anomalie nella gestione dei file random.

Riempimento dei campi

Proseguiamo oltre andando a vedere come si riempiono effettivamente i vari campi del record.

Supponiamo perciò che vogliamo inserire nel campo C\$, relativo al cognome e lungo 20 byte, la voce "GARIBALDI".

Dato che tale cognome è lungo 9 caratteri si avranno 11 byte inutilizzati: si può scegliere di avere questi 11 spazi vuoti a destra o a sinistra del cognome, o meglio si può decidere se "giustificare" tale cognome a sinistra oppure a destra (rispettivamente).

Nei due casi, supponendo che "GARIBALDI" si trovi nella stringa A I\$, avremo LSET C\$ = A1\$ (giustificazione a sinistra) oppure

RSET C\$ = A1\$ (giustificazione a destra)

Riempiti così i vari campi con opportune stringhe, non ci rimarrà altro che andare a scrivere sul disco il record così formato: ecco che avremo ora a che fare con il faticoso "numero del record".

Infatti scrivendo ad esempio PUT 1,5, andremo a scrivere il nostro record al 5° posto nel file "Rubrica".

Evidentemente possiamo scrivere PUT I,N, dove N è stavolta una variabile, che magari impostiamo noi da tastiera con una istruzione INPUT.

Viceversa se scriviamo PUT 1 semplicemente, ordiniamo di scrivere il record "subito dopo" al record precedentemente scritto.

Questo fatto di poter stabilire noi dove collocare il record all'interno del file è evi-

FILES RANDOM (Scrittura e Lettura)	
Funzione logica	Istruzione dell'MBASIC
1 - Apertura del file	OPEN "R", [#] buffer, "nomefile" [, lunghezza]
2 - Allocazione spazio del buffer per i record	FIELD [#] buffer, lungh.campo AS var.stringa [, ...]
3 - Riempimento del buffer con giustific. a sinistra o destra	LSET var.stringa = stringa RSET var.stringa = stringa
4 - Conversione variabili intere singola precisione doppia precisione in stringhe	var.stringa = MKI\$(var.intera) var.stringa = MKS\$(var.reale) var.stringa = MKD\$(var.doppia precis.)
5 - Scrittura dati dal buffer al disco	PUT [#] buffer [,numero record]
6 - Memorizzazione del record nel buffer	GET [#] buffer [,numero record]
7 - Conversione di stringhe in variabili intere variabili singola precisione variabili doppia precisione	var.intera = CUI(stringa di 2 byte) var.sing.prec. = CVS(stringa di 4 byte) var.dop.prec. = CVD(stringa di 8 byte)
8 - Valore del puntatore al record successivo	LOC(buffer)
9 - Chiusura del file	CLOSE [buffer] [, ...]

Tabella 1 - Corrispondenza tra le varie funzioni logiche di gestione dei file random e le funzioni dell'MBasic.

dentemente il maggior pregio dei file random nei confronti di quelli sequenziali.

Letture dei record

Sempre all'interno dello "stesso" programma possiamo andare a leggere un certo record, avente un suo numero d'ordine.

Ad esempio se vogliamo leggere il record n. 20, non abbiamo altro da fare che impostare l'istruzione

```
GET 1,20
```

Anche in questo caso valgono le stesse considerazioni fatte per la "PUT" riguardo al numero del record che può essere un numero, una variabile, oppure non comparire nemmeno nell'istruzione nel qual caso si fa riferimento al record successivo a quello utilizzato l'ultima volta.

Di colpo, con la nostra GET 1,20 andiamo a leggere "tutto" il record ed a questo punto avremo le nostre brave variabili C\$, N\$, I\$ e T\$ che conterranno qualcosa.

Volendo leggere l'indirizzo basterà evidentemente impostare:

```
PRINT I$
```

come pure si potrà trasferire I\$ dentro un'altra variabile con un'ovvia istruzione:

```
!!$ = I$
```

Viceversa per non vedere generalmente dei caratteri grafici, lettere strane, ecc, "non" consigliamo di scrivere

```
PRINT NS !!
```

Per vedere invece il vero contenuto (si tratta infatti della codifica di un numero) dobbiamo usare le istruzioni CVI, CVS e CVD, analoghe alle varie "MKX\$". Dovremo però sapere se la stringa proveniva da un valore intero reale o in doppia precisione, pena ulteriori malfunzionamenti.

Nel nostro caso dovremo ad esempio impostare

```
PRINT CVS (NS)
```

Come si sarà già notato, nel caso dei random, non si deve più fare i conti con virgole, virgolette e punti e virgola, il che è veramente notevole. L'ultima istruzione relativa ai file random è la LOC (1) che ci dice qual è il valore attuale del puntatore ai record del file: ricordiamo che tale valore è quello a cui "andremo" ad accedere in lettura o in scrittura e perciò incrementato di uno rispetto ad un precedente accesso.

Ci eravamo dimenticati di chiudere il file alla fine! Presto fatto: una "CLOSE" ci permetterà questa azione, come era facilmente prevedibile.

Considerazioni finali

Siamo ormai in grado di poter giudicare, in base al nostro problema, quale struttura usare: random o sequenziale. Possiamo a questo punto ricordare le note salienti, i "pro" e i "contro" dei nostri file: faremo i nostri bravi conticini in termini di numero dei record, lunghezza dei campi, necessità

di tempi di accesso più o meno brevi, necessità più o meno frequente di modificare i campi di taluni record, effettiva necessità di un "puntatore" (indice numerico) dei record.

Il tutto lo dovremo tradurre da un lato (file sequenziali) con una struttura elastica, flessibile e semplice, ma interamente "sequenziale".

Dall'altro lato avremo una struttura statica (ricordate la disquisizione sulla pessima traduzione del termine file con "flusso"?!), rigidamente prestabilita ma ideale per l'accesso rapido qua e là all'interno del file stesso. Può spaventare, lo ammettiamo, il fatto di dover separare i vari campi con opportuni "separatori" nel caso dei sequenziali, ma certo non è che il rigore formale dei random sia da meno...

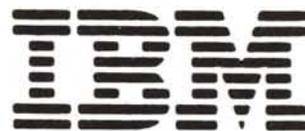
Se per leggere il decimo elemento, nel primo caso, dobbiamo leggerne altri nove, nel secondo caso dobbiamo portarci appresso la struttura del record, accedere al record stesso e isolare la parte desiderata, eventualmente sfruttando le conversioni a variabili numeriche. Ma attenti a non sbagliare! Non aggiungiamo altro, se non il consiglio di provare, per il proprio problema, ad usare entrambi i tipi di file, per poter decidere, in base ai criteri su-esposti, qual è il migliore. **MC**



COMPUTER SYSTEMS



apple computer



COMMODORE

olivetti

ROMA - Via G. Lanza 101-103-105 (tra Via Merulana e Via Cavour) Tel. 738224-738854

M fermata Vittorio Emanuele (linea A) - Via Cavour (linea B)

OSTIA LIDO - Via A. Carabelli 108-110-112
(tra Via Isole Capoverde e Via delle Azorre)

Tel. 5697686

GROTTAFERRATA - Via Trento 32-34 (Centro dimostrativo)
Sabato aperto anche pomeriggio



Elenco del software disponibile su cassetta o minifloppy

Per ovviare alle difficoltà incontrate da molti lettori nella digitazione dei listati pubblicati nelle varie rubriche di software sulla rivista, MCmicrocomputer mette a disposizione i programmi più significativi direttamente su supporto magnetico. Riepiloghiamo qui a fianco i programmi disponibili per le varie macchine, ricordando che i titoli non sono previsti per computer diversi da quelli indicati. Il numero della rivista su cui viene descritto ciascun programma è riportato nell'apposita colonna; consigliamo gli interessati di procurarsi i relativi numeri arretrati, eventualmente rivolgendosi al nostro Servizio Arretrati utilizzando il tagliando pubblicato in fondo alla rivista.

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Valsolda 135, 00141 Roma.

Le cassette utilizzate sono Basf C-60 Compusette II; i minifloppy sono Basf singola faccia singola densità.

Codice	Titolo programma	MC n.	Prezzo	Note
=====				
APPLE II				
DA2/00	Shape Tablet	22	15000	!
DA2/01	Motomuro	26	15000	!
DA2/02	&DEBUG	28	15000	!
DA2/03	EDIT + INPUT	29	15000	!
DA2/04	Basic modulare	34	15000	!
DA2/05	ANNA Animation Lang.	35/37	15000	!
DA2/06	Miniset + Leva-DOS	37	15000	!
=====				
COMMODORE 64				
C64/01	Briscola	25	17000	!
C64/02	Serpentone	29	17000	!
C64/03	Othello	29	17000	!
C64/04	Chase	33	17000	!
C64/05	Spreadsheet	34	30000	!
C64/06	Bilancio familiare	35	17000	!
C64/07	The dark wood	36	17000	!
C64/08	Totocalcio: sis.rido	37	17000	!
C64/09	Orchetes	37	17000	!
D64/01	Spreadsheet	34	15000	!
=====				
COMMODORE VIC-20				
CVC/01	VIC-Maze	19	17000	! Config. base
CVC/02	Pic-Man	23	17000	! Config. base
CVC/03	Briscola	25	17000	! Config. base
CVC/04	Grand Prix	28	17000	! Config. base
CVC/05	Frogger	26	17000	! RAM: almeno + 3 K
CVC/06	Invaders	29	23000	! RAM: + 16 K
CVC/07	Othello	29	17000	! RAM: + 16 K
CVC/08	SKI	31	17000	! Config. base
CVC/09	VIC-quiz	32	17000	! RAM: almeno + 8 K
CVC/10	Zigurat	33	17000	! Config. base
CVC/11	Extended Basic	36	17000	! RAM: + 16 K
CVC/12	Fireman	36	17000	! Config. base
DVC/01	EXMA	27/28	15000	! RAM: + 16 K
=====				
SINCLAIR SPECTRUM				
CSS/01	TRILAB	28	17000	!
CSS/02	SET di caratteri	27/29	17000	!
CSS/03	Grafica TREDIM	29	17000	!
CSS/04	Ippica	30	17000	!
CSS/05	Graphic-Comp	32	17000	!
CSS/06	Macchina del tempo	34	17000	!
CSS/07	Piramide di Iunnuh	35	17000	!
CSS/08	Over Basic	37	17000	!
=====				
TEXAS TI-99/4A				
CT9/01	Macchina del tempo	27	17000	!
CT9/02	Simon	29	17000	!
CT9/03	Babilonia	30	17000	!
CT9/04	Labirinto 3D	31	17000	!
CT9/05	Piramide di Iunnuh	33	17000	! Extended Basic
CT9/06	Scrabble	34	17000	!
CT9/07	Morphy	35	17000	!
CT9/08	Equo canone	37	17000	!
=====				

Nota:

l'iniziale del codice e' C per le cassette, D per i minifloppy