

Easy-Format

di William Patti - Pisa

Tutti coloro che hanno un floppy disk drive, anche gli utenti muniti solo di registratore (datassette), che si stanno accingendo a comprarne uno, sapranno certamente che per formattare un disco nuovo si impartisce il comando: OPEN 15,8,15, "N0:NOME-DISCO":CLOSE 15.

L'ID cioè l'identificatore è formato da due lettere, e non è possibile usare simboli semi-grafici; con questo sistema è possibile utilizzarli e vediamo come: OPEN 15,8,15, "N0:NOME-DISCO"+CHR\$(34)+"ID":CLOSE 15.

Al posto di ID potrete inserire due caratteri semi-grafici a vostro piacere, attenti a non dimenticare la virgola prima dei due caratteri, pena il non funzionamento del trucchetto.

Certamente saprete che, ci sono due tipi di formattazione. Quella «lunga» si impartisce per un dischetto nuovo da utilizzare in seguito; il drive ronzerà per una trentina di secondi, e avrete così il dischetto pronto per l'uso; (anche se ci sono in commercio programmi in L.M. i cosiddetti FAST FORMAT, che formattano dischetti in 15 secondi). Quella «corta» serve per cancellare tutti i file memorizzati su disco, il drive ronzerà per un paio di secondi con un notevole risparmio di tempo e di logorio del drive ed avrete il dischetto pronto per essere riutilizzato.

Funzionamento del programma

Dato il RUN apparirà il seguente messaggio in reverse:

EASY-FORMAT ATTIVARE CON: SYS49152

Impartite SYS49152 e premete RETURN, apparirà:

INSERT A DISKETTE BEFORE FORMATTING OPERATION
PRESS RETURN WHEN READY...

Questo programma scritto in L.M. (caricatore Basic) per C64/128, permette

È disponibile, presso la redazione, il disco con i programmi presentati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 263.

di formattare un dischetto nuovo senza immettere il Nome del Disco e l'ID, in quanto sono stati inseriti nel programma, utilizzando la routine del S.O. Kernal «CHROUT» che scrive il carattere in OUTPUT.

A questo punto non dovete far altro che inserire un dischetto nuovo nel drive, e premere RETURN, per dare inizio alla formattazione (formattazione lunga) alla fine apparirà il messaggio dello STATUS del disco, se tutto è andato bene, dopo la formattazione apparirà il messaggio 00,OK,00,00 (ho provveduto ad inserire nel programma una subroutine di stampa dei messaggi di errore del drive), alla fine avrete un dischetto che si chiamerà UTILITY e avrà come ID: MC (come MCmicrocomputer).

Editor 160

di Gaetano Minardi - Niscemi (CL)

Appena passato al Commodore 128, una delle prime cose che notai, fu l'ampliamento dell'editor di linea, che rispetto al C64 risultava raddoppiato. Infatti permetteva di inserire quattro righe di programma per un totale di ben 160 colonne. Questa utile innovazione mi lasciò inizialmente di stucco; solo dopo capii che era una delle caratteristiche

peculiari di questa rivoluzionaria macchina, perché passando in modo 80 colonne le linee visualizzabili tornavano a due, ma i caratteri visualizzabili restavano sempre $(80 \times 2) = 160$.

Affascinato dalle potenzialità del C 128 programavo costantemente col Basic 7.0, ma di tanto in tanto passavo in modo 64 e mi accorgevo, con mio grosso rammarico, di dovere dimenticare (oltre alle numerose istruzioni che il nuovo Basic offriva) i 160 caratteri in una sola riga di programma Basic. Avendoci fatta ormai l'abitudine, incappavo frequentemente in fastidiosi messaggi d'errore che lasciavano intravedere i difetti, o meglio i limiti, del Commodore 64 (da me definito comunque il padre degli home computer). So intanto per esperienza che questi bug (se così possono essere chiamati) sono una scelta della Commodore, la quale ha preferito lasciare libera quanto più memoria possibile, piuttosto che migliorare e rendere più semplice la programmazione in Basic.

Così ho deciso di rendere più flessibile il Basic, per avvicinarlo a quello del C 128. A tal proposito ho creato tante espansioni ma anche tanti programmi di utilità per sfruttare meglio le caratteristiche e i pregi del C64.

Editor 160 è un utilissimo programma (realizzato come di consueto in linguaggio macchina) che permette anche al C

Easy-Format

```

10 REM EASY-FORMAT PER C-64/128
20 REM WRITTEN BY PATTI WILLIAM - PISOFT <C> 1989
30 REM PER MC/MICROCOMPUTER
40 FORJ=0TO224:READS:POKE49152+J,S:CK-CK+S:NEXT
50 IF CK<>25208 THEN PRINT"ERRORE NEI DATA ":END
60 PRINT"(CLR)(RVS) EASY-FORMAT ATTIVARE CON: SYS49152 ":NEW
70 DATA 169,147,32,210,255,169,29,32,210
80 DATA 255,169,138,160,192,32,30,171
90 DATA 32,228,255,201,13,208,249,32,118
100 DATA 192,32,201,255,169,78,32,210,255
110 DATA 169,58,32,210,255,169,85,32,210,255
120 DATA 169,84,32,210,255,169,73,32,210,255
130 DATA 169,76,32,210,255,169,73,32,210,255
140 DATA 169,84,32,210,255,169,89,32,210,255
150 DATA 169,44,32,210,255,169,77,32,210,255
160 DATA 169,67,32,210,255,32,118,192
170 DATA 32,198,255,32,207,255,32,210,255
180 DATA 32,183,255,201,64,208,243
190 DATA 169,13,32,210,255,32,231,255,96
200 DATA 169,2,162,8,160,15,32,186,255
210 DATA 169,0,32,189,255,32,192,255,162
220 DATA 2,96,13,13,13,29,73,78,83,69,82,84
230 DATA 32,65,32,68,73,83,75,69,84,84,69
240 DATA 32,66,69,70,79,82,69,32,70,79,82
250 DATA 77,65,84,84,73,78,71,13,29,79,80
260 DATA 69,82,65,84,73,79,78,13,13,13,29
270 DATA 18,80,82,69,83,83,32,18,82,69,84
280 DATA 85,82,78,32,87,72,69,78,32,82,69
290 DATA 65,68,89,46,46,46,13,13,29,29,0

```

READY.

```

10 REM *****
15 REM *
20 REM *          EDITOR 160
25 REM *
30 REM *          DI
35 REM *
40 REM *          MINARDI GAETANO
45 REM *
50 REM *          VIA SERBATOIO N. 44
55 REM *
60 REM *          93015-NISCEMI (CL)
65 REM *
70 REM *          TEL. 0933/952002
75 REM *
80 REM *****
100 I=49152
110 READA:IFA<OTHEN160
120 POKEI,A:I=I+1:GOTO110
130 REM
140 REM ATTIVAZIONE PROGRAMMA
150 REM RICORDARSI PRIMA DI SALVARLO
155 REM
160 SYS (49152):NEW
170 REM
180 REM DATA PROGRAMMA
190 REM
200 DATA162,000,138,157,000,008,232
210 DATA208,250,162,224,134,252,169
220 DATA000,133,251,160,000,177,251
230 DATA145,251,200,208,249,232,208
240 DATA238,169,159,141,194,230,141
250 DATA255,231,169,004,141,162,232
260 DATA141,180,232,162,160,134,252
270 DATA169,000,133,251,160,000,177
280 DATA251,145,251,200,208,249,232
290 DATA224,192,208,236,162,007,142
300 DATA019,165,142,022,165,142,036
310 DATA165,142,205,165,142,208,165
320 DATA142,241,165,142,011,166,142
330 DATA210,170,142,062,172,142,213
340 DATA171,232,142,245,164,142,107
350 DATA165,142,132,165,142,186,165
360 DATA142,231,165,142,006,166,142
370 DATA206,170,142,236,171,142,058
380 DATA172,142,149,171,142,154,171
390 DATA142,182,167,142,188,180,162
400 DATA168,142,110,165,169,167,141
410 DATA024,003,169,002,141,025,003
420 DATA173,032,208,072,162,255,160
430 DATA255,140,032,208,136,208,250
440 DATA202,208,245,104,141,032,208
450 DATA162,000,189,023,193,157,031
460 DATA165,232,224,011,208,245,162
470 DATA000,189,219,192,157,167,002
480 DATA232,224,060,208,245,169,009
490 DATA133,044,169,001,133,043,234
500 DATA234,234,169,053,133,001,076
510 DATA034,193,072,138,072,152,072
520 DATA169,127,141,013,221,172,013
530 DATA221,048,042,032,002,253,208
540 DATA003,108,002,128,032,188,246
550 DATA032,225,255,208,026,032,021
560 DATA253,032,163,253,169,053,133
570 DATA001,032,024,229,169,167,141
580 DATA024,003,169,002,141,025,003
590 DATA108,002,160,076,114,254,164
600 DATA011,185,252,007,145,095,136
610 DATA208,248,234,169,000,141,217
620 DATA236,169,005,141,218,236,169
630 DATA000,141,053,229,141,000,009
640 DATA032,068,166,169,008,141,164
650 DATA225,076,250,192,000,000,000
660 DATA-1

```

READY.

(2048). In questo modo ho potuto creare comodamente un nuovo buffer di input di ben 160 caratteri (quanto richiesto dalle nuove condizioni), che ovviamente va da \$0800 (2048) a &08A0 (2208).

Appena lanciato, il programma oltre ad effettuare tutte le operazioni citate in precedenza, provvede a copiare il Kernal e l'Interprete Basic in Ram, in modo da potere modificare alcuni vettori. Così ho memorizzato la routine di RESTORE a partire dalla locazione \$02A7 fino a \$02E3. Queste non devono essere alterate per nessun motivo, perché provocherebbero immancabilmente il blocco totale del sistema. Un'ultima cosa: dopo che il programma viene eseguito la zona di memoria in cui esso è memorizzato, cioè \$C000, può di nuovo essere usata per altri scopi, in quanto avendo effettuato tutte le operazioni necessarie al funzionamento non occorre più tenere il programma ancora in memoria.

Una tecnica di Overlay

di Roberto Morassi - Pistoia

È noto che la gestione dei puntatori del C64 non permette (o meglio, lo permette solo a certe condizioni) quello che in altri sistemi si chiama «chaining», cioè il caricamento e lancio di un programma Basic da parte di un altro, condividendone le variabili già acquisite. Vediamo il perché e come si può risolvere il problema tramite una routine in L.M.

Il LOAD da programma differisce da quello da tastiera in tre punti: non aggiorna il puntatore di fine programma, non esegue il CLR delle variabili e fa partire il prog-2 in autorun. Quest'ultimo potrà quindi funzionare, condividendo le variabili già dichiarate dal prog-1 (Overlay), solamente se è più corto: in caso contrario, verrà caricato «sopra» le variabili preesistenti cancellandole e l'Overlay non sarà possibile.

C'è però un altro aspetto, forse meno evidente. Supponiamo di voler far partire comunque il prog-2 con le sue proprie variabili, cioè indipendentemente da quelle precedenti (per esempio, inserendovi un CLR proprio all'inizio). Ciò potrebbe servire in molti casi: una serie di programmi che si richiamano a vicenda dal disco, o uno molto lungo da dividere in due parti, o ancora un programma-

menu che ne fa partire un altro a scelta...

Provate pure... e vedrete che anche questo NON è possibile, se il prog-2 è più lungo! Il perché è evidente: il puntatore di fine programma \$2D-2E (che segna l'inizio della zona-variabili) «punta» sempre alla fine del prog-1, e non appena il prog-2 dichiarerà una variabile questa verrà allocata proprio nel «corpo» del programma stesso, alterandolo e mandandolo prima o poi in blocco.

CHAIN è una breve utility in L.M. che consente l'Overlay indipendentemente dalla lunghezza del nuovo programma (naturalmente senza condivisione di variabili!).

Essa inserisce un «wedge» nella routine del S.O. che interpreta i comandi Basic (\$A7E4) e che viene richiamata non appena il prog-2 parte: il wedge aggiorna il puntatore \$2D-2E ed esegue il CLR, per cui il nuovo programma funzionerà normalmente. Si assume che la zona Basic sia quella di default, a partire da \$0801 (#2049).

La routine deve essere caricata in memoria dal prog-1, proprio all'inizio: l'allocazione qui suggerita è da \$02A8 (#680), ma può essere rilocata in qualunque altra zona senza modifiche. Il caricamento del prog-2 dovrà essere fatto non con un LOAD, ma con la sintassi seguente (il nome del programma può essere sostituito, se occorre, da una variabile-stringa):

```
SYS680 "PROG-2",8
```

Il prog-2 potrà a sua volta caricare un prog-3 con la stessa istruzione, e così via: ovviamente, il wedge continuerà a funzionare solo se la sua zona di memoria non viene utilizzata per caricarvi altri dati.

Come esempio applicativo della routine CHAIN, vi propongo un menu automatico (SUPERMENU) per la scelta, caricamento e auto-run di un programma da disco. Tali menu sono essenzialmente di due tipi:

(1) universali: leggono la directory del disco e propongono i singoli file per il caricamento, dietro pressione di un tasto o input di un numero;

(2) personalizzati: non leggono la directory, ma vanno integrati con i nomi dei programmi prima di registrarli sul disco, e aggiornati in caso di variazioni.

SUPERMENU appartiene a questo secondo tipo. Per installarlo, procedete come segue:

(a) scrivete il nome del disco nella varia-

64 di usare 160 caratteri in una sola linea di programma. La routine viene memorizzata a partire da \$C000 (49152), quindi per attivarlo basta la sola SYS49152. Per comodità ho dovuto spostare l'area Basic a \$0900 (2304) riducendola di 256 byte, infatti prima partiva da \$0800

bile ND\$ della linea 90.

(b) Scrivete i nomi esatti dei programmi che volete caricare, sotto forma di DATA, nelle linee 270-290. Ad esempio:

```
270 DATA PIPPO,COPIATUTTO,PACMAN,
ecc.
```

Potete scrivere fino a 26 nomi. Se queste linee non bastano, potete intercalarne delle altre ma facendo attenzione a non inserire DATA oltre la linea 300, che NON va assolutamente modificata né cancellata perché contiene il «flag» di fine lista.

(c) Inserite il disco e date RUN 1000 che salverà il programma col nome MENU. Per i successivi aggiornamenti, ricaricate il MENU, aggiungete o modificate i DATA e date RUN 500 che sostituirà la nuova versione alla precedente.

(d) Se possedete una utility di ordinamento della directory, spostate il MENU all'inizio in modo che si possa caricare anche con il comando LOAD «*»,8.

Dando il RUN la routine di «chaining» verrà installata in memoria, e comparirà la schermata con l'indice dei programmi: per caricarne uno, basterà premere il tasto con la relativa lettera.

Purtroppo, non è detto che tutti programmi funzionino immediatamente: ci sono quelli in L.M. che richiedono un successivo SYS, quelli già in auto-run che richiedono un caricamento particolare, ecc.

In questi casi si usano in genere dei brevi programmi Basic, detti «loader» o «boot», che servono appunto a caricare e far partire il programma principale. Vediamo alcune possibilità.

(1) Programmi Basic normali o compi-

```
5 REM ** LOADER 1 - R.MORASSI **
10 IFX=0THENX=1:LOAD"NOMEDELPROGRAMMA",8,1
20 SYS00000
```

READY.

```
5 REM ** LOADER 2 - R.MORASSI **
10 POKE53280,7:POKE53281,7:M$="ATTENDERE, PREGO !"
20 PRINT"(CLR)(BLK)"TAB(255)SPC(245-LEN(M$)/2)M$
30 PRINT"(YEL)(HOME)LOAD"CHR$(34)"NOMEDELPROGRAMMA"CHR$(34)",8,1:"
35 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)PF780,0:PF781,0:PF782,0:SYS00000"
40 POKE198,3:POKE631,19:POKE632,13:POKE633,13:END
```

READY.

```
5 REM ** LOADER 3 - R.MORASSI **
10 SYS65409:POKE646,PEEK(53281)
20 PRINT"(HOME)LOAD"CHR$(34)"NOMEDELPROGRAMMA"CHR$(34)",8,1:"
30 POKE631,19:POKE632,13:POKE198,2:END
```

READY.

Overlay

```
5 REM ** SUPERMENU - BY R.MORASSI **
6 :
10 FORX=0TO47:READY:POKE680+X,Y:NEXT
20 DATA 165,020,024,105,020,141,008,003
30 DATA 165,021,105,000,141,009,003,104
40 DATA 104,076,104,225,056,169,004,229
50 DATA 122,169,008,229,123,144,014,165
60 DATA 174,133,045,165,175,133,046,032
70 DATA 096,166,032,083,228,076,228,167
80 :
90 AS="*****":ND$="NOMEDELDISCO"
100 ABS="(BLK) _____"
110 ACS="(BLK) _____"
120 DIMF$(26):SYS65409:POKE53280,0
130 READF$:IFF$<"**"THENNF=NF+1:F$(NF)=F$:GOTO130
140 PRINT"(CLR)(YEL)"AS"_____":AS:
150 PRINTAS"|"TAB(60-LEN(ND$)/2)ND$:TAB(71)|"AS:
160 PRINTAS"_____":AS
170 PRINT:PRINTAB$
180 FORX=1TO13
190 PRINT"|(WHT)"CHR$(64+X)"(BLK)|(CYN)"F$(X):
200 PRINTTAB(19)"(BLK)|(WHT)"CHR$(77+X)"(BLK)|(CYN)"F$(X+13):TAB(38)"(BLK)|"
210 NEXT:PRINTACS:PRINT"(DOWN)(WHT) "
220 PRINT"(LEFT)?":FORX=1TO10:NEXT:PRINT"(LEFT) ":FORX=1TO10:NEXT
230 GETN$:IFN$<CHR$(65)ORN$>CHR$(NF+64)THEN220
240 FC=ASC(N$)-64:PRINTTAB(19-(LEN(F$(FC)/2)))(RVS)"F$(FC) "
250 POKE53280,14:SYS680F$(FC).8,1
260 END
270 DATA ?????????????
280 DATA ?????????????
290 DATA ?????????????
300 DATA *
310 END
320 :
500 OPEN15,8,15,"I":PRINT#15,"S0:MENU":CLOSE15
1000 OPEN15,8,15,"I":CLOSE15:SAVE"MENU",8:VERIFY"MENU",8

READY.
```

lati (con caricatore Basic incorporato). Si possono inserire tal quali nel menu: il wedge CHAIN li farà partire normalmente.

(2) Programmi in L.M. da caricare al di fuori della zona Basic con LOAD «PROG»,8,1, e richiedono un SYS di attivazione. Supponiamo di avere un programma PIPPO che si carica a partire dalla locazione 4096 e si attiva con SYS 17571. Se necessario, rinominate il programma sul disco, facendolo ad esem-

pio precedere da una freccia o altro simbolo, o incorporandovi il numero del SYS (PIPO17571). Caricate il LOADER 1 qui riportato, sostituendo «PIPPO17571» nella linea 10, e «17571» nel SYS della linea 20. Salvate il loader sul disco col nome PIPPO e inserite quest'ultimo nel menu. L'auto-run farà partire la riga 10 che caricherà PIPPO17571 senza sovrapporsi al loader: quest'ultimo ripartirà dopo il LOAD ma questa volta con X=1, per cui la riga 10 verrà saltata e la riga 20 darà il SYS richiesto.

(3) Programmi in L.M. che vanno nella zona Basic (da 2048 in poi), cancellando il loader: si dovrà usare il buffer di tastiera sia per il LOAD che per il SYS. Caricate LOADER 2 e inserite, come prima, il nome del programma (modificato) e il relativo SYS nella riga 30. Al posto del messaggio di attesa M\$ potete mettere il nome del programma, o altre istruzioni. Registrate infine il loader con il nome che riporterete nel menu.

(4) Programmi auto-run che partono da soli se caricati con LOAD «PROG»,8,1 (in locazioni antecedenti la zona-schermo). Usate il LOADER 3 con le stesse modalità di prima.

(5) Ci sono infine dei programmi che richiedono comunque un intervento manuale, ad esempio la pressione di RUN STOP + RESTORE eventualmente eseguita da un SYS in modo diretto. Si può usare il LOADER 2, aggiungendo nella schermata (metà inferiore dello schermo) le necessarie istruzioni.

Elenco del software disponibile su supporto magnetico

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 sotto 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 Carlo Perrier 9, 00157 Roma.

Codice	Titolo Programma	MC n.	Prezzo
COMMODORE AMIGA			
DAM/01	F-15	63	15000
DAM/02	Gest. liste programmi	64	15000
DAM/03	Studio di Funzioni	66	15000
DAM/04	Math Pack	68	15000
DAM/05	Redcode & Mars (Core Wars)	68	15000
DAM/06	Life	69	15000
DAM/07	Rubrica Telefonica	70	15000
DAM/08	Piramidi	70	15000
DAM/09	Regolazione dei colori	71	15000
DAM/10	Analitica	71	15000
DAM/11	Grafici	72	15000
DAM/12	Traduttore	73	15000
DAM/13	La Borsa	74	15000
DAM/14	DMA Music Compiler	74	15000
DAM/15	Poker	78	15000
DAM/16	Programmi per il Copper	79	15000
DAM/17	Mandelbrot mania	81	15000
DAM/18	SF-Search File	86	15000
DAM/19	AMipac	87	15000
DAM/20	Click	93	15000
DAM/21	Editor di Videogame	95	15000
DAM/22	RandomLand	98	15000
MS-DOS			
DMS/01	Pictter + Morse	67	15000
DMS/02	Melole + Spam	68	15000
DMS/03	Pretty + Scritte scorrevoli + Compute	69	15000
DMS/04	Emulatore CGA per Hercules	70	15000
DMS/05	Turbo Directory	71	15000
DMS/06	Math Tool S	72	15000
DMS/07	Biortmi + Routine	72	15000
DMS/08	Salvavideo + Scritte scorrev. + PG151	73	15000
DMS/09	Optimizer + Indenter dBase III	74	15000
DMS/10	Joystick Controller	74	15000
DMS/11	BootSlow + SlowDown + Turbo Utility	76	15000
DMS/12	Redcode & Mars (Core Wars)	76	15000
DMS/13	Gestione Errori Critici Disco + PosCur	77	15000
DMS/14	Finestre & Desk	78	15000
DMS/15	General Manager	78	15000
DMS/16	Tool DS	79	15000
DMS/17	Pulidown Menu + Retrace	80	15000
DMS/18	Righe	81	15000
DMS/19	La spada di Krail	82	15000
DMS/20	Regressione	82	15000
DMS/21	Tesseract + Charset Editor	83	15000
DMS/22	Sega-File + Installatore	84	15000
DMS/23	Shape Tool-kit	87	15000
DMS/24	Starfox	88	15000
DMS/25	Inline in Turbo C + Svuota Keyboard Buffer	89	15000
DMS/26	Sistemi parametrici + GDFIR 1.0	90	15000
DMS/27	QuickMatrix + Stampa Virtuale	91	15000
DMS/28	Whereis + Bold.C + Setmenu	92	15000
DMS/29	Bobo + MGA 100	93	15000
DMS/30	War + LS (2 dischi)	94	20000
DMS/31	Warbot Arena	95	15000
DMS/32	Cmouse + Sp-Build + Tratt.Lap	96	15000
DMS/33	Il diavoleto di Maxwell + Criminals (2 dischi)	97	20000
DMS/34	Num-Unlock & ESC + Math Parser	98	15000
ATARI ST			
DST/01	Virus Killer	74	15000
DST/02	Mandelbrot + Prolez. Ort. + Bilancio	78	15000
DST/03	Diagrammi di Henon	81	15000
DST/04	Paroliamo	84	15000
DST/05	Enalotto	85	15000
DST/06	Cianche	83	15000
DST/07	Flow-Chart	87	15000
DST/08	Funz. Manual	89	15000
DST/09	RedCode	88	15000
DST/10	Pattern Generator 2.1	92	15000
DST/11	ST Hopalong + Moto casuale	93	15000
DST/12	Speedy Maid	94	15000
DST/13	Masod	94	15000
DST/14	Designer	95	15000
DST/15	Pro-Function 1.0	96	15000
DST/16	Tris e Carene	97	15000
DST/17	Masod + Print + Multitas + Memmon	98	15000
COMMODORE 128			
D28/01	MMCalc	53	15000
D28/02	Hardcopy 128	55	15000
D28/03	Sheet It	57	15000
D28/04	Star Quest	58	15000
D28/05	Family Budget	60	15000
D28/06	La Casa Stregata	61	15000
D28/07	Strutture 80/33	63	15000
D28/08	Bas 80 V. 2.0a	64	15000
D28/09	Paint 80 1.0	65	15000
D28/10	Bas 80 V. 2.11	66	15000
D28/11	Calendario Perpetuo + Montecarlo	67	15000
D28/12	Disegna Circuiti	68	15000
D28/13	Mark's Data Base	70	15000

Codice	Titolo Programma	MC n.	Prezzo
D28/14	Label Disk + Disk Editor + Dem DOS	71	15000
D28/15	Pulidown 128HR + Menu + Drawer	72	15000
D28/16	Prospettive	73	15000
D28/17	Char 80 V. 1.0	74	15000
D28/18	Italia 128	75	15000
D28/19	Super Sprite	77	15000
D28/20	Othello	80	15000
D28/21	Expert System Shell 128	81	15000
D28/22	Kit di programmazione S.O.G.A.R. 128	82	15000
D28/23	Caratteri Programmabili	83	15000
D28/24	Bush & Mouse	84	15000
D28/25	Astrolabio	89	15000
D28/26	Sort Vettore Stringa + Input Stringa	90	15000
D28/27	Piano Cartesiano + Calendario	91	15000
D28/28	Bingol 90	92	15000
D28/29	Kosla View + Clock 128 + File Data Maker	93	15000
D28/30	Curve Polari	94	15000
D28/31	Biblio 128	95	15000
COMMODORE 64			
D64/14	Link-64	57	15000
D64/15	New Char 2.2	58	15000
D64/16	Music 64	59	15000
D64/17	TRX-MEM	59	15000
D64/18	WOS + WBasic	60	15000
D64/19	Strange Basic + Dracula	63	15000
D64/20	File Rescue	64	15000
D64/21	La Casa	64	15000
D64/22	Digital Voice	65	15000
D64/23	Vita 3D	65	15000
D64/24	Corso di Linguistica	66	15000
D64/25	Archipius	66	15000
D64/26	Math Pack Plus	66	15000
D64/27	Scroll + Multitask + Classifica	67	15000
D64/28	Calend. Perpetuo + Eletto Telecamera	68	15000
D64/29	Listing Plus + Utility Data	69	15000
D64/31	Trucchi e Routine per programmatori	71	15000
D64/32	Flow-Chart + Flower's Love	73	15000
D64/33	Sprite Editor	76	15000
D64/34	Portfolio 64 + Elimin. bordi schermo	77	15000
D64/37	Intonatore	81	15000
D64/38	Gendata 64	82	15000
D64/40	Block Utility	84	15000
D64/41	Sprite 6499	85	15000
D64/42	Mot Mot Labyrinth + Tool 64	86	15000
D64/43	Titolatore 64	88	15000
D64/44	Editor Musicale	89	15000
D64/45	Dump Variabili	90	15000
D64/46	FBasic V1.0 + Tod-Basic	91	15000
D64/47	Le sedici carte + Stampa Directory	92	15000
D64/48	REL-Arch 64. Statistica 3.0	94	15000
D64/49	MegaGame 64	95	15000
D64/50	PikPok	96	15000
D64/51	Sam: Scritte scorrevoli	97	15000
D64/52	Overlay + Easy Format + Editor 160	98	15000
MSX			
DMX/01	Toto 13	60	15000
DMX/02	Painter	62	15000
DMX/03	MSX Bank	63	15000
DMX/04	Grafica 3D + Hard Copy	65	15000
DMX/05	Easy Disk	66	15000
DMX/06	Classifiche	67	15000
DMX/07	Magic Paint	67	15000
DMX/08	Autogest	68	15000
DMX/09	Compiler v. 1.01	69	15000
DMX/10	Diskmap	70	15000
DMX/11	Mini dBase MSX	71	15000
DMX/12	Grafica in Turbo Pascal	72	15000
DMX/13	Math Pack Plus 3.20	73	15000
DMX/14	RGB CAD	75	15000
DMX/15	Simple Desk	76	15000
DMX/16	The MSX2 Super Print	77	15000
DMX/17	Grafica in Turbo Pascal (Graph 1&2)	77	15000
DMX/18	Hard Copy	78	15000
DMX/19	HEXJUMP	79	15000
DMX/20	Utilities in Turbo Pascal	80	15000
DMX/21	dBase MSX Plus	81	15000
DMX/23	PutChar + 4010 S	84	15000
DMX/24	Analysis + Lister Basic	85	15000
DMX/25	Aliment MSX	86	15000
DMX/26	SortDir + Hectogonus + 3D Mover	87	15000
DMX/27	Tool/Extended Basic + Bindata Bas	88	15000
DMX/28	Chain & Window + Chain Utility	90	15000
DMX/29	Latino Pack	91	15000
DMX/30	Anoid	97	15000
DMX/30	MSX-DOS Utility	98	15000

Nota:
Il supporto è rappresentato da dischi da 5.25" e 3.5" in relazione alle caratteristiche dei disk drive presenti sul computer in oggetto.