

software COMMODORE 64



Questo mese vi presentiamo tre utility molto interessanti e stimolanti che toccano degli aspetti che non mancheranno di suscitare la vostra curiosità. Con le prime due sarà possibile stampare caratteri giganti su carta mentre, con l'ultima, avremo... due computer in uno. Introduciamole brevemente.

La prima si chiama **MAXI** e ci mette a disposizione sullo schermo una finestra video 11×10 , dotata di tutti i comandi di edit, nella quale potremo comporre un testo che ci verrà restituito su carta ingrandito.

La seconda ha nome **MAXI PRINT** e si differenzia dalla prima per il fatto che non ci saranno più messi a disposizione caratteri fissi bensì di grandezza programmabile (fino ad un solo carattere che occupa tutto il foglio). E non solo. Il nostro maxi testo potrà essere stampato sia in orizzontale che in verticale, tipo striscione (l'autore ce ne ha mandato uno di un metro di lunghezza). Con questa seconda utility, non comporremo più il testo in una finestra video, ma scriveremo la stringa che vogliamo come risposta ad una richiesta di input convenzionale, introducendo anche la grandezza desiderata per i caratteri: penserà poi il computer a calcolare se i valori impostati sono coerenti, cioè se, con quella determinata grandezza di carattere, la stringa entra nel foglio. Questo naturalmente per la stampa in orizzontale perché, per quella in verticale chi... più carta ha, più ne metta. Questi primi due programmi, oltre che sulle stampanti indicate dagli autori, hanno funzionato anche sulla vecchia GP100-VC e sulla nuova MPS 803.

Infine, con la terza utility, chiamata **SPLIT RAM**, avremo la possibilità di tenere in memoria due programmi contemporaneamente ed eseguire, indifferentemente, l'uno o l'altro, abilitandolo con un semplice comando che potremo dare sia in modo diretto che da programma.

Buon divertimento.

Maxi

di Antonio Panico - Lecce

Questo programma è un editor di maxi caratteri su stampante che permette di ottenere, stampato su carta e con caratteri giganti, un testo precedentemente composto su video. Con **MAXI**, un foglio di carta per stampante da 66 righe e 80 colonne

viene trasformato in un foglio da 11 righe da 10 maxi caratteri per riga. Ciascun carattere è contenuto in una griglia di 8×8 maxi pixel — ciascuno rappresentato, a scelta, da uno dei simboli presenti sulla tastiera — che riproduce esattamente la struttura del carattere corrispondente contenuto nella Rom generatrice di caratteri del 64.

Uso del programma

Dopo aver ricopiato attentamente il listato e dato il < Run >, appare sullo scher-

mo, decentrata verso destra, una finestra che riproduce un foglio di carta per stampante (vedi figura 1, pag 142) delimitato da due tratteggi. La scritta nella parte inferiore del video ci avvisa che siamo in ambiente "scrittura testo" e il cursore lampeggia nell'angolo in alto a sinistra della finestra nella quale potremo comporre il testo che poi sarà riprodotto ingrandito sulla carta. La composizione si avvale di una routine di input controllato che ci dà la possibilità di scrivere utilizzando tutte le funzioni di edit della macchina. È quindi possibile spostarsi nelle quattro direzioni utilizzando i tasti cursore, cancellare con < Del >, inserire caratteri con < Inst >, spostarsi nell'angolo in alto a sinistra della finestra con < Home > e così via. Inoltre, il programma è stato concepito in modo che, nella finestra, sia abilitato lo scrolling verso l'alto e verso il basso della scritta, ottenuto sempre con i tasti funzione, che in ogni modo non permette al testo di fuoriuscire dalla zona di edit. In altre parole, quando una riga di testo raggiunge ad esempio il margine inferiore della finestra, si arresta e la stessa cosa avviene nello scrolling verso l'alto.

```

100 REM -----
120 REM EDITOR DI MAXI-CARATTERI
150 REM PER COMMODORE 64
190 REM DI ANTONIO PANICO - LECCE
230 REM -----
899 :
900 POKE53280,6:POKE53281,6:PRINT"███";DIMV$(23):G$=""
910 PRINTTAB(24) " I: I: PRINTTAB(24) "-----"
920 FORI=0TO10:PRINTTAB(24-LEN(STR$(I)))I:PRINTTAB(24) "NEXTI
930 PRINTTAB(24) "-----" :PRINTTAB(24) " I: I:
940 V$="" :FORW=0TO23:V$(W)=LEFT$(V$,W+1):NEXT
950 S3$="" :S1$=S3$+S3$+S3$+S3$:S2$=LEFT$(S1$,24):S4$=LEFT$(S3$,8)
970 PRINT"███ STAMPA MAXI-CARATTERI ███TAB(38)███ ███V$(14)TAB(38)███ ███";
1000 F1%=0:F2%=0:F3%=0
1200 PRINTV$(14)███ SCRITTURA TESTO ███
1210 PRINTV$(17)███ PREMERE █ SHIFT & RETURN █ AL TERMINE █ DEL TESTO"
1220 PRINTV$(20)███ █ @ █ ALL' INIZIO DI UNA RIGA PROVOCA ███;
1230 PRINT" L'ARRESTO DELLA STAMPA IN QUEL PUNTO."
1240 PRINTV$(14)███ SCRITTURA IN CORSO ███;IFF2%=1THEN1280
1260 FORI=0TO10:T$(I)=S3$:NEXTI:GOSUB4000
1280 HZ=0:KZ=0:XZ=26
1290 VZ=HZ+2:T$=T$(HZ):GOSUB3000:T$(HZ)=T$:IFR%=-1THEN1370
1300 IFR%=1ANDHZ>0THENHZ=HZ-1:GOTO1290
1310 IFR%=2ANDHZ<10THENHZ=HZ+1:GOTO1290
1320 IFR%=1ANDHZ=0ANDT$(0)=S3$THENGOSUB3800:GOTO1290
1330 IFR%=2ANDHZ=10ANDT$(0)=S3$THENGOSUB3900:GOTO1290
1340 IFR%=10ORF2%=2THEN1290
1350 IFR%=3THEN1280
1360 IFR%=4THEN1260
1370 PRINTV$(17)S1$S1$V$(20)S1$S1$V$(20);
1400 PRINTV$(17)"CORREZIONI ? (S/N) ███";
1410 GETR$:IFR%<"S"ANDR%<"N"THEN1410
1420 PRINTV$(17)S1$:IFR%="S"THENF2%=1:GOTO1200
2000 PRINTV$(14)███ STAMPA ███;IFF3%=1THEN2042
2010 PRINTV$(16)"CARATTERE USATO NELLA":PRINTV$(17)"STAMPA SU CARTA : ███";
2020 PRINTV$(19)███ CONFERMI ? (S/N) ███;
2025 GETR$:IFR%<"S"ANDR%<"N"THEN2025
2030 PRINTR$:IFR%="S"THEN2040
2032 PRINTV$(21)"CARATTERE DA USARE : ███";
2034 GETG$:IFG%<CHR$(32)OR(G%<CHR$(127)ANDG%<CHR$(160))THEN2034
2036 PRINTG$:FORN=1TO250:NEXT
2040 PRINTV$(16)S1$S1$V$(19)S1$V$(21)S1$
2042 PRINTV$(16)███ PREPARARE LA STAMPANTE E POSIZIONARE ███;
2044 PRINT" LA CARTA FACENDO COINCIDERE IL TRAT- ███";
2046 PRINT" TEGGIO COL BORDO SUPERIORE DELLA T- STINA."
2048 PRINTV$(21)"BATTERE. █ C █ A OPERAZIONE CONCLUSA : ███";
2050 GETR$:IFR%<"C"THEN2050
2055 PRINTR$:IFF1%=1THEN2300
2060 PRINTV$(16)DECODIFICA IN CORSO: RIGA";
2070 F1%=1:SZ=10:FORI=0TO10:PRINTV$(I),I:FORK=0TO7:C$(I,K)="" :NEXTK:T$=T$(I)
2080 IFT$=S3$THEN2180
2090 IFLEFT$(T$,1)=""@THENSZ=I-1:GOTO2200
2100 IFRIGHT$(T$,1)="" THENT$=LEFT$(T$,LEN(T$)-1):GOTO2100
2120 LS=1130+I*40:FORJ=1TOLEN(T$):A%=PEEK(LS+J-1)-256*(PEEK(53272)=23)

```


tribuendo ad ogni 0 ricavato uno spazio e ad ogni 1 un carattere predefinito G\$: si ottengono così otto stringhe di otto caratteri. A questo punto, sommando ordinatamente a queste stringhe le otto del carattere che nel testo rappresentano il successivo carattere per ogni linea T\$(i), si ottengono le otto C(k,i).

Nella linea 2070, per ogni riga I di testo, le C(i,k) vengono inizializzate a stringhe vuote ("") mentre la T\$(i) trattata e depositata in T\$. Se quest'ultima è una sequenza di spazi (riga vuota), le relative C(k,i) rimangono stringhe vuote (2080). Successivamente vengono eliminati gli spazi terminali della T\$(2100).

Nella linea 2120 viene calcolata la locazione di schermo del primo carattere dalla T\$(variabile LS) ed è letto il suo contenuto (A%) che ci fornisce il D/Code del carattere. Nella 2150 è invece calcolato l'indirizzo della prima delle otto locazioni della Rom dei caratteri (MC) che contengono la struttura del carattere avente come D/Code A%. Continuando, nella linea 2160 vengono letti dalla Rom, uno per uno, gli otto valori decimali che ci interessano (C%) ciascuno dei quali è trasformato in binario ottenendo C\$(subroutine 3700). Infine, le C\$(i,k) saranno composte dalla somma delle C\$ successive relative ai caratteri della riga di testo.

Come importante nota aggiungiamo che, per operare in questo modo, prima di leggere la Rom dei caratteri occorre disattivare il timer della tastiera e sospendere l'I/O rispettivamente con

POKE 56334, PEEK (56334) AND 254
POKE 1, PEEK (1) AND 251
vedi la linea 2150.

Dopo aver letto la Rom, le normali condizioni vengono ripristinate con:

POKE (1) PEEK (1) OR 4
POKE 56334, PEEK (56334) OR 1
dove la prima Poke setta a zero il bit 2 attivando la Rom dei caratteri e la seconda setta ad 1 lo stesso bit (dell'altra locazione) attivando il CIA prima disattivato.

Esame del listato

900-1000 : inizializzazioni e disegno parte fissa della videata; G\$ è il carattere elementare di cui si è parlato;
1200-1240 : scritte fisse sul video;
1260 : inizializzazione delle 11 linee del testo (matrice T\$(i), i=0-10)
1280-1370 : gestione composizione testo sul video;
1400-1420 : conferme testo o correzioni;
2010-2036 : conferme o cambiamento carattere elementare G\$;

2040-2055 : attesa preparazione stampante e carta;

2060-2180 : lettura testo scritto sul foglio video e sua decodifica, cioè trasformazione di ogni riga di testo T\$(i) in stringhe che stampate su carta formano il maxi testo. Ogni T\$(i) dà luogo ad otto C\$(i,k): sequenze di spazi alternati a caratteri elementari G\$;

2200-2240 : eliminazione blank terminali delle C\$(i,k);

2300-2330 : stampa del testo con maxi caratteri;

2400-2470 : menu opzioni di continuazione;
2500 : fine programma.

Subroutine

3000-3210 : input controllato del testo;
3700-3740 : conversione decimale binario: lo zero è sostituito da un blank (" ") e l'uno da un carattere G\$;

3800-3900 : scrolling del testo nel foglio video;
4000 : scrittura del testo sul foglio video.

Maxi Print

di Vincenzo Garofalo - Cosenza

Con questo programma è possibile stampare su carta, ingrandita a piacimento, in orizzontale o verticale (tipo striscione) una qualsiasi riga composta sullo schermo.

Appena dato il <Run>, viene presentato il menu mediante il quale sceglieremo l'opzione di stampa che preferiamo, cioè orizzontale o verticale. In fase di selezione ci verranno richiesti anche altri parametri: se scegliamo l'opzione di stampa orizzontale, verrà richiesta l'altezza e la larghezza di ciascun maxi carattere ed ogni unità introdotta rappresenta il grado di ingran-



Figura 1 - La finestra in cui viene composto il testo nel programma "MAXI".

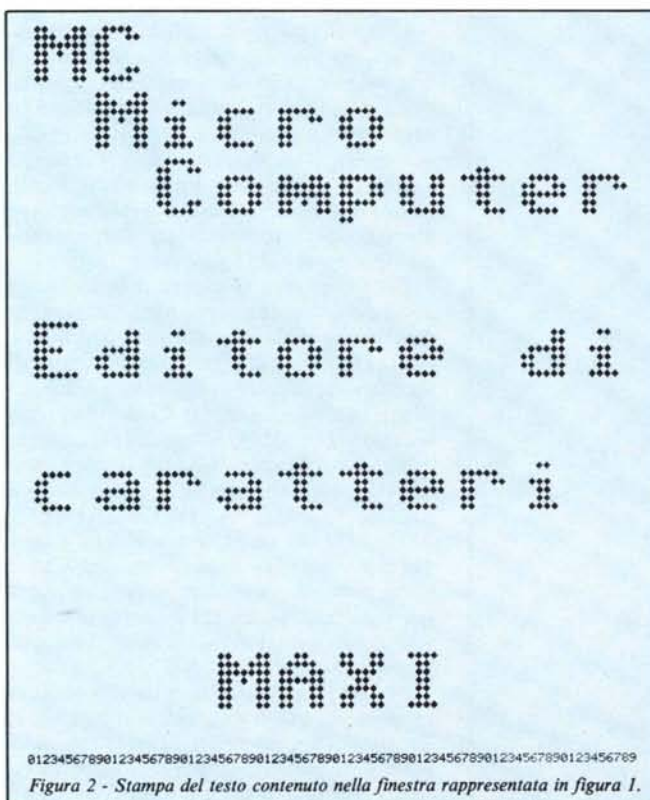


Figura 2 - Stampa del testo contenuto nella finestra rappresentata in figura 1.

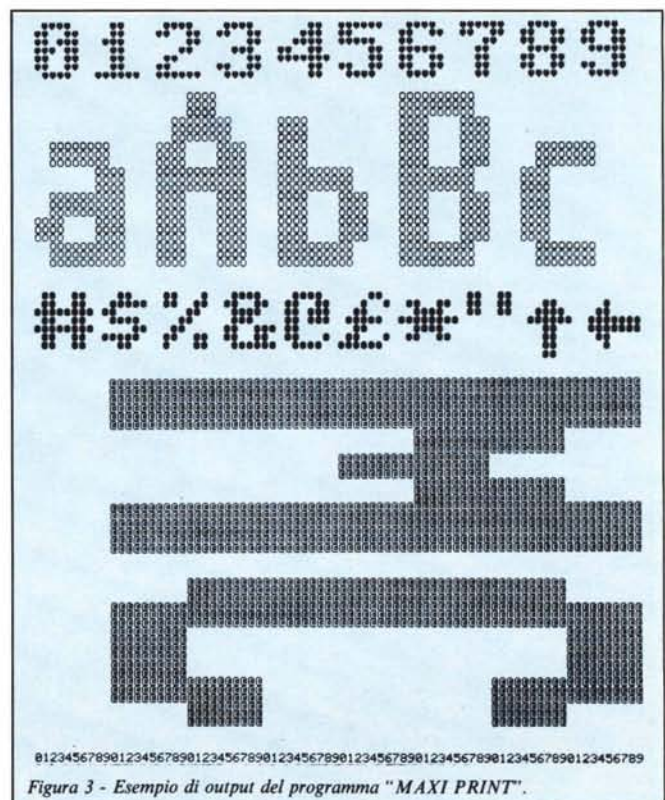


Figura 3 - Esempio di output del programma "MAXI PRINT".


```

10 rem -----
20 rem -
30 rem - maxi Print Per commodore 64 -
60 rem -
70 rem - vincenzo garofalo - cosenza -
80 rem -
90 rem -----
100 Poke53280,6:Poke53281,14
110 Print"***** The Maxi Print *****"
120 Print" (C) 1985 by Vincenzo Garofalo"
130 Print" Fasi di attivita':"
140 Print" 1.....Stampa in orizzontale"
150 Print" 2.....Stampa in verticale"
160 Print" 3.....Fine lavoro"
170 gotoP$:ifOP$<"1"orOP$>"3"then170
180 ifOP$="3"thenPrint:end
190 input"Stringa da stampare";a$
200 input"Stringa di base";bs$
210 input"Larghezza caratteri";l
220 ifOP$="2"then250
230 input"Altezza caratteri";h
240 nc=10/l;ifnc<len(a$)thenPrint"Stringa troppo
lunga":fort=1to3000:next:goto190
250 Print"tab(255);tab(252)"Attendere Pre90"
260 dimb$(len(a$),8)
270 fora=1tolen(a$)
280 as=asc(mid$(a$,a,1)):gosub540
290 forb=0to7
300 Poke56333,3:Poke1,51:P=Peek(55296+as#8+b)

```

```

310 Poke1,55:Poke56333,131
320 k=128
330 x=kandp
340 ifxthenb$(a,b+1)=b$(a,b+1)+bs$:goto360
350 b$(a,b+1)=b$(a,b+1)+" "
360 k=k/2:ifk=>1then330
370 nextb,a
380 open1,4:ifOP$="2"then470
390 fora=1to8
400 forb=1toh:Print#1,chr$(15);
410 forc=1tolen(a$)
420 ford=1to8
430 fore=1tol
440 Print#1,mid$(b$(c,a),d,1):nexte,d,c
450 Print#1,chr$(8):nextb,a
460 Print#1,chr$(15):close1:run
470 fora=1tolen(a$)
480 forb=1to8
490 forc=1to1:Print#1,chr$(15);
500 ford=8to1step-1
510 fore=1to10
520 Print#1,mid$(b$(a,d),b,1):nexte,d:Print#1,
chr$(8):nextc,b,a
530 Print#1,chr$(15):close1:run
540 ifas>63andas<96thenas=as-64
550 ifas>95andas<128thenas=as-32
560 ifas>128thenas=as-128
570 return

```

dimento di ciascuna griglia elementare 8x8 (compresi i blank) contenente il carattere. Per la stampa in verticale, viene chiesta la larghezza del carattere in quanto l'altezza è fissata ad un ingrandimento di 10 volte. Analizziamo ora alcuni punti importanti del programma.

Alla linea 240 si controlla la lunghezza della stringa da stampare in funzione della larghezza dei caratteri mentre nelle linee 300 e 310 viene prelevato al generatore di caratteri il byte che sarà convertito in binario e, successivamente, stampato. La prima poke spegne gli impulsi di interrupt e la seconda sgancia il Vic II lasciando "visibile" il generatore di caratteri. La conversione in binario avviene alle linee 320-360 e la subroutine 540 effettua una conversione da codice Ascii a codice di schermo.

Il programma gira perfettamente con le stampanti modello 801 e 803 della Commodore; comunque, "eventuali" punti critici, per quanto riguarda altre stampanti, sono racchiusi tra le linee 400, 450, 460, 490 e 520. I codici 15 e 8 inviati alla stampante evitano che essa, tra una riga e l'altra, ne lasci una vuota.

Si noti che durante la copiatura del programma è bene fare attenzione alla linea 240 che contiene più di 80 caratteri e digitarla usando il metodo di abbreviazione delle parole chiave.

Elenco variabili

- l : larghezza caratteri
- h : altezza caratteri
- nc : numero caratteri che possono entrare in una riga
- as : codice Ascii del singolo carattere della stringa da stampare
- k : contatore per la conversione in binario
- x : cifra binaria utile alla conversione

- a : utilizzata tre volte. Prima per la conversione in binario; poi per la stampa rispettivamente in orizzontale e in verticale
- b : come a
- c,d,e : variabili di conteggio per la stampa
- a\$: stringa principale da stampare
- b\$() : matrice contenente i valori binari
- bs\$: stringa che nella stampa rappresenta il singolo punto
- op\$: scelta del tipo di stampa: 1 = orizzontale; 2 = verticale

Nota

Ripartiamo il listato in questo programma stampato con il set maiuscolo/minuscolo per facilitarne la copiatura. L'unico punto oscuro può essere forse rappresentato dalle lettere "n" ed "h" che trovate in campo inverso nella linea 110. Esse equivalgono rispettivamente, a chr\$(14), che abilita il set alternativo di caratteri ed a chr\$(8) che disabilita l'effetto della pres-

sione contemporanea di Shift e del logo Commodore. Essi possono essere scritti come nella linea 110.

Split ram

di Andrea Sandri - Pisa

Quando si stanno elaborando dei dati ed è necessario l'uso contemporaneo di più programmi (ma anche in altre occasioni), una delle cose più noiose è senz'altro l'impossibilità di avere in memoria almeno due programmi e di poterli usare indifferentemente, magari utilizzando un semplice comando che permetta di saltare dall'uno all'altro senza dover ricorrere a continui e snervanti caricamenti.

La semplice utility proposta dà una soluzione a questo problema.

Con essa la memoria del C 64 viene divisa in due aree (una di 23 ed un'altra di 15K) completamente indipendenti tra loro e nelle quali è possibile caricare separatamente due diversi programmi con cui lavorare alternativamente. Per un comodo funzionamento del tutto, è utile avere una semplice istruzione Basic che permetta di saltare, sia in modo diretto che da programma, da un'area di memoria all'altra. A questo scopo viene sacrificato il normale funzionamento del comando LET, del resto quasi mai utilizzato con la sua vera funzione nei programmi, rendendolo così disponibile per abilitare il passaggio da una zona di memoria ad un'altra.

Riassumendo, dopo il <Run>, tramite il comando LET, proposto in una nuova veste, potremo passare da una zona all'altra della memoria in cui risiedono i due programmi ed il cambiamento di situazione verrà evidenziato da un cambiamento del colore del bordo del teleschermo. MC

```

10 REM -----
20 REM          SPLIT RAM
30 REM          DI ANDREA SANDRI
40 REM          56100 PISA
50 REM -----
100 POKE33280,0:POKE33281,0:PRINT"*****"
110 POKE56,160:POKE25681,0:POKE25682,0
120 POKE44,0:POKE33282,3:POKE33283,100
130 CLR:POKE1,55
140 PRINT"*****ATTENDERE PREGO"
400 FOR I=40960 TO 49152:POKE I,PEEK(I):NEXT
600 FOR I=53000 TO 53115:READ A:POKE I,A:NEXT
610 POKE1,54:POKE40988,7:POKE40989,207
620 PRINT"*****HO FINITO I"
630 CLR:LET LET=NEW:END
1000 DATA16, 56, 165, 44, 233, 8, 200
1010 DATA54, 165, 45, 141,208, 207,165
1020 DATA46,141,209, 207,169,100, 133
1030 DATA44, 169, 0, 141, 0, 100, 133
1040 DATA51, 173, 210, 207, 133, 47,133
1050 DATA45,133, 49, 173, 211, 207, 133
1060 DATA50, 133, 46, 133, 40, 169, 160
1070 DATA133, 56, 133, 32, 169, 5, 141
1080 DATA32, 208, 32, 94, 166, 96, 165
1090 DATA45,141, 210, 207, 165, 46, 141
1100 DATA211, 207, 169, 0, 133, 44, 169
1110 DATA 0, 141, 0, 0, 133, 51, 173
1120 DATA208,207, 133, 47, 133, 43, 133
1130 DATA49,173, 209, 207, 133, 46, 133
1140 DATA40, 133, 50, 169, 99, 133, 56
1150 DATA133, 52, 169, 6, 141, 32, 208
1160 DATA 32, 94, 166, 96

```




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 Carlo Perrier 9, 00157 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	!
DA2/07	27 programmi grafici	38	30000	!
DA2/08	Adventure Editor	38	15000	!
DA2/09	Animazione funzioni	42	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.rid.	37	17000	!
C64/09	Orchetes	37	17000	!
C64/10	Wordprocessor	38	17000	!
C64/11	Helicopt	38	17000	!
C64/12	Finestra grafica	39	17000	!
C64/13	Paroliamo	39	17000	!
C64/14	Scarabeo	40	17000	!
C64/15	Magazzino	41	17000	!
D64/01	Spreadsheet	34	15000	!
D64/02	ADP Basic	da 35 a 39	15000	!
D64/03	Wordprocessor	38	15000	!
D64/04	Paroliamo	39	15000	!
D64/05	Data base Galileo	40/41	15000	!
D64/06	Magazzino	41	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
CVC/13	Accordi per chitarra	39	17000	! RAM: almeno + K
CVC/14	Piramide di Iunnuh	39	17000	! RAM: almeno + K
CVC/15	Il castello	40	17000	! RAM: + 16 K
DVC/01	EXMA	27/28	15000	! RAM: + 16 K
=====				
MSX				
CMX/01	Sound editor	42	17000	!
=====				
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	!
CSS/09	Prospettiva	38	17000	!
CSS/10	Motomuro	39	17000	! 48 K RAM
CSS/11	Othello	40	17000	!
CSS/12	The dark wood	40	17000	!
CSS/13	Musica	41	17000	!
CSS/14	Calcolo matriciale	42	17000	!
CSS/15	Database	42	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	!
CT9/09	Scopa	39	17000	!
CT9/10	Montecarlo	39	17000	! Extended Basic
CT9/11	Totocalcio	41	30000	!
=====				
Nota:				
l'iniziale del codice e' C per le cassette, D per i minifloppy				
=====				