

commodore
COMPUTER CLUB

N.7

Lire 2000

Dicembre 1983 - Sped. Abb. Post. gr.III/70 - Suppl. a Computer n 62 - Distr. Me. Pe. Spa

Mensile per gli utenti di sistemi Commodore

**Due regali eccezionali:
Un assembler
per tutti i Commodore
e la maschera - memo
del programmatore**

**Giochi e programmi
per il VIC
ed il 64**



espande all'infinito la tua esperienza

HES

REBIT
COMPUTER
A DIVISION OF G.B.C.



software a misura d'uomo

MEE

Tecnologia del domani



La MEE **esclusivista** per l'Italia dei prodotti **VERBATIM** **DATALIFE** Vi propone oltre alla più completa gamma di supporti magnetici anche armadi ignifughi Rosengrens, nastri inchiostriati per tutte le stampanti, mobili e tavoli porta terminali per l'arredamento dei CED. Con i prodotti MEE le registrazioni dei Vostri dati non hanno più problemi di affidabilità.

MEEMEEM
Vi prego voler inviare la vostra completa documentazione relativa a:

- Dischetti e inoltre
- Disk-packs
- Nastri inchiostriati
- Armadi ignifughi
- Mobili per CED



MEE - Memorie per Elaboratori Elettronici s.r.l.
Forniture per Centri Elaborazione Dati
Sede Amm.va: 20144 Milano - Via Boni, 29
Tel. 4988541 (4 linee r.a.) 4986296-4984196
Filiali e Agenzie: Milano - Bergamo - Torino - Biella - Padova - Parma - Bologna - Firenze - Ancona - Roma - Napoli - Catania - Oristano - Bari - Genova - Bolzano - Mestre -

EUMMEUMMEUM

ragione sociale _____
nome _____
via _____
Cap. _____ Città _____

C commodore

a Roma

è



P.le Asia, 21 - 00144 ROMA EUR - Tel. (06) 5916438

CENTRO REGIONALE di
DISTRIBUZIONE, VENDITA, ASSISTENZA

- Vic. 20 e accessori
- Dischi Winchester (commodore) :5, 8, 22, 30 MB
- Backup su nastro magnetico
- Sviluppo procedure personalizzate
- Contratti di manutenzione per parti di ricambio



SOMMARIO

	REMARKS	Vic 20	Commodore64	Sistemi	Generali
	<i>Edipo</i>				
07	Può un computer imparare dai propri errori?	•	•	•	
	<i>Speciale Natale/1</i>				
12	Un assembler per tutti i Commodore	•	•	•	
	<i>Didattica</i>				
23	Imparare a programmare col VIC (7ma dispensa)	•			•
	<i>Speciale Natale/2</i>				
28	Maschera-memo del programmatore	•	•		
	<i>Un programma musicale</i>				
33	Tastiera	•			
	<i>Guida ai videogames</i>				
36	Il controllo del cursore	•			•
	<i>Sistemi</i>				
42	Ed ora passo al Pet			•	
	<i>Scrittura personalizzata</i>				
46	Stampa in corsivo	•	•	•	
53	Guida mercato Commodore				•
24	Come collaborare a Computer Club				•

Commodore Computer Club - mensile indipendente per gli utenti di sistemi Commodore

Direttore responsabile: Michele di Pisa

Redazione: Alessandro De Simone

Direzione, redazione: V.le Famagosta, 75 - 20142 Milano - Tel. 02 / 8467348

Pubblicità: Milano - Micro Croce, Paola Bevilacqua, Gianluigi Centurelli, Tina Ronchetti, Villa Claudio - V.le Famagosta, 75 - 20142 Milano - Tel. 02 / 8467348/9/40

Prezzi e abbonamenti: Prezzo per una copia Lire 2.000

Arretrati il doppio. Abbonamento per dieci fascicoli lire 18.000

Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti) L. 34.000. I versamenti vanno indirizzati a

Minisystem - Italia s.r.l., mediante assegno bancario, vaglia o utilizzando il c/c postale n. 11909207.

Composizioni: Minisystem - Italia

Selezioni: Org. A. G.

Stampa: La Litografica s.r.l. - Busto A.

Registrazione: Tribunale di Milano n. 2/10/1982 - Sped. in abb. post. gr.

III n. 70 quale supplemento alla rivista Computer - Pubb. inferiore al 70%

Con i nuovi programmi italiani della Sirius Elettronica, Commodore 64 serve proprio a tutto.

Per tutti i possessori, anche futuri, del personal computer Commodore 64 c'è un'importante novità: la creazione di una vasta serie di nuovi programmi realizzati espressamente per le esigenze italiane.

Con questi programmi i tecnici italiani della Sirius Elettronica mettono a disposizione un servizio che trova Utenti nei più diversi settori delle attività produttive: dal dentista all'albergo, dal negozio al teatro. E molti altri ancora.

Programmi semplici da usare, chiari, e razionali. E in italiano.

Tutto questo per valorizzare sempre di più Commodore 64.

PROGRAMMI SU DISCO PER CBM - 64

	Prezzo al pubblico (IVA esclusa)
GESTIONE AGENTI RAPPRESENTANTI	120.000
GESTIONE AMMORTAMENTI MUTUI	60.000
GESTIONE ANAGRAFICHE	120.000
GESTIONE SCADENZE POLIZZE	300.000
GESTIONE AGENZIA VIAGGI	250.000
TOTOCALCIO (SVILUPPO COLONNARE)	100.000
GESTIONE CONTI CASA	100.000
GESTIONE CONDOMINIO (SEMPLIFICATO)	200.000
IMPARIAMO IL BASIC	100.000
DICHIARAZIONE I.V.A.	150.000
DISTINTA BASE	-80.000
GESTIONE EQUO CANONE	100.000
MODELLO 740 ORDINARIO	100.000
MODELLO 740 NORMALE	150.000
GESTIONE STAMPA FATTURA	80.000
CENTO PROGRAMMI PER CBM-64 (DIDATTICO)	80.000
GESTIONE FIDO CLIENTI	120.000
GESTIONE CONTO CORRENTE	150.000
GESTIONE CONTI	150.000
GESTIONE APPUNTAMENTI	150.000
GESTIONE ORDINI	150.000
GESTIONE CANTINE	250.000
GESTIONE I.V.A. SEMPLIFICATA	60.000
GESTIONE CLUBS	250.000
GESTIONE OFFICINE	250.000
GESTIONE PORTAFOGLIO CLI/FOR	200.000
GESTIONE TEATRO	120.000
GESTIONE LIBRERIE E BIBLIOTECHE	150.000
GESTIONE RISTORANTI	380.000
MAILING LIST	150.000
WORD PROCESSOR	160.000
WORD ON MAIL	250.000
SCADENZIARIO EFFETTI CLI/FOR	200.000

Prezzo al pubblico (IVA esclusa)

GESTIONE PULIZIA CAPI	200.000
CONTABILITÀ FATTURE C/F	200.000
CONTABILITÀ MONOAZIENDALE SEMPLIFICATA	400.000
GESTIONE ALBERGHI	400.000
GESTIONE PARRUCCHIERI	400.000
GESTIONE GOMMISTI	400.000
LEGGE 373 (ISOLAMENTI TERMICI)	150.000
COMPUTO METRICO	400.000
INGEGNERIA CIVILE I	250.000
INGEGNERIA CIVILE II	250.000
REVISIONE PREZZI	300.000
MAGAZZINO E FATTURAZ. AGGANCIATE	400.000
CONTABILITÀ GENERALE	400.000
GESTIONE OTTICI	300.000
GESTIONE DENTISTI	300.000
GESTIONE FARMACIE	400.000
GESTIONE STUDI MEDICI	300.000
GESTIONE CONDOMINIO ADVANCED	400.000
DUPLICATORE ARCHIVI	400.000
SINTETIZZATORE MUSICALE	120.000
DATA BASE	200.000

PROGRAMMA APPLICATIVI SU CASSETTA PER VIC-20

SIR-MAG. 8	62.500
SIR-MAG. 16	87.000
SIR-MAG. 32	98.000
RICETTARIO DI CUCINA	38.000
BIORITMO	41.000
PROGRAMMA PUBBLICITARIO PER NEGOZI	41.000
INDIRIZZARIO	41.000
SIR-CONT - CONTABILITÀ DOMESTICA	49.000
PROGRAMMI GIOCO SU CASSETTA	24.500
GORTEK AND MICROCHIPS	24.500

Desidero conoscere il punto vendita, a me più vicino, in cui ottenere informazioni sui programmi applicativi della Sirius Elettronica.

Nome _____

Cognome _____

Via _____

CAP _____ Città _____

Inviare a:
Sirius Elettronica
Via Imperia, 21 - 20142 Milano
Tel. 02/8490400-8431368

SIRIUS elettronica

Può un computer imparare dai propri sbagli?

Un piccolo esempio di intelligenza artificiale.

QUESTO che presentiamo è uno dei programmi che fanno molto riflettere sulle possibilità di un calcolatore. Si dimostra, infatti, che, attraverso una serie di domande e risposte, un computer può "apprendere" informazioni e fornirle quando si richiedono.

Una volta che il lettore digita il programma (che gira su qualsiasi computer Commodore, compreso il Vic 20 senza espansioni), può provarne il funzionamento, rispettando la seguente procedura riportata a solo titolo di esempio (in maiuscolo ciò che il lettore deve digitare, in minuscolo i messaggi del computer).

Digita il primo nome della serie

PASQUALE

e quando sei pronto premi un tasto

Premere un tasto qualunque.

Pasquale. Ho indovinato (s/n)?

N

Che cos'era?

SANDRO

Che domanda faresti parlando di Sandro?

PORTA GLI OCCHIALI

Che risposta daresti (s/n)?

S

Vuoi giocare ancora (s/n)?

S

Pensa a ciò che devo indovinare...

e quando sei pronto premi un tasto

Porta gli occhiali?

N

Pasquale. Ho indovinato (s/n)?

N

Cos'era?

ANNA

Che domanda faresti parlando di Anna?

E' intelligente

Che risposta daresti (s/n)?

N

Eccetera. Come si può notare il computer, all'inizio, ha a disposizione solo un nome (Pasquale) e chiede subito se il nominativo da indovinare è proprio Pasquale oppure no. In caso negativo gli si comunica il nuovo nome (Anna), che verrà memorizzato insieme alla sua caratteristica peculiare. All'inizio il gioco è monotono, ma diventa divertente dopo l'inserimento di una ventina di nominativi, che possono anche esser ripetuti inserendo ogni volta una caratteristica diversa. Per esempio, dapprima Pasquale sarà solo

```

10 REM *** APPRENDIMENTO AUTOMATICO PER COMPUTER ***
20 REM ***                                     ***
30 REM *** PER QUALSIASI COMPUTER (ALMENO CON SK)***
40 REM ***                                     ***
50 REM ***PASQUALE D'URIA VIA E. DE FILIPPIS.145***
60 REM ***      34013 CAVA DEI TIRRENI (SA)      ***
70 REM ***      TEL. 089/492617                    ***
80 REM ***                                     ***
100 DIM A$(300), A$(150): B$ = "(S/N/NO)"
110 PRINT "LEGGI DA NASTRO?" : GOTO 160
120 GOSUB 500 : IF B$ = "" OR B$ <> "S" AND B$ <> "N" THEN 120
130 IF B$ = "N" THEN 610
140 OPEN 1:1:0
150 J=0
160 INPUT#1,A$(J):INPUT#1,A$(J)
170 IF A$(J) <> "FINE" THEN J=J+1: GOTO 160
180 CLOSE 1
190 I=0: PRINT "PENSA A CIÒ CHE DEVO INDOVINARE..."
200 PRINT "...E QUANDO SEI PRONTO PREMI UN TASTO." : GOSUB 600
210 IF ASC (A$(I)) = 64 THEN 270
220 PRINT A$(I) "?"

```

SE VUOI ESSERE DI SCEGLIERE

Ogni giorno in Europa si apre un nuovo computer shop. Un pubblico sempre più numeroso è attirato verso il personal e si rivolge ai negozi specializzati per trovare la sua marca preferita.

Tu che hai capito qual è il futuro dei computer e hai deciso di aprire un negozio, cerchi un nome che dia prestigio e una organizzazione che non ponga vincoli ma offra vantaggi concreti.

Computeria vuol dire negozi di computer fin dal 1979.

Computeria è anche una organizzazione che ha avviato rapporti di collaborazione con tutti i principali fornitori, perciò i suoi affiliati possono scegliere e vendere le marche più prestigiose e richieste.

E inoltre Computeria ti dà un prezioso know-how, una ricchissima dotazione di programmi, supersconti esclusivi, vantaggi economici sul leasing.

E tanta pubblicità.

Se vuoi essere libero di scegliere quello che vuoi vendere nel tuo negozio, l'organizzazione Computeria è la tua scelta obbligata.

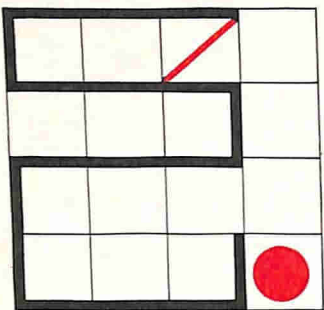


 **COMPUTERIA®**

La catena senza catene.

RE LIBERO





SOFTGAMES

OFFRI ARGOMENTI AL TUO COMPUTER!

CATALOGO INVERNO '83/84

LEGENDA: C=CASSETTA
R = CARTUCCIA • D = DISCO

		PREZZI AL NETTO DI IVA
GIOCHI DI AZIONE		
CHOPLIFTER	R	84.000
DAVID'S MIDNIGHT MAGIC	D	67.000
FROGGER	C	67.000
JAWBREAKER	R	68.000
LODE RUNNER	D	69.000
PROTECTOR	C	69.000
SHAMUS	D	67.000
SHAMUS II	C	69.000
JUMPMAN	C	77.000
JUMPMAN JUNIOR	R	77.000
PITSTOP	R	77.000
LUNAR OUTPOST	R	77.000
CRASH CRUMBLE & CHOMP	C	59.000
DIDATTICI		
HEY DIDDLE DIDDLE	D	56.000
SONG WRITER	D	
DELTA DRAWING	R	
UTILITIES		
QUICK BROWN FOX	D	149.000
SIMON BASIC	D	105.000
LOGOMOTION	D	

		PREZZI AL NETTO DI IVA
AVVENTURA		
WITNESS	D	97.000
TEMPLE OF APSHAI	R	77.000
UPPER REACHES OF APSHAI	C	37.000
COURSE OF RA	C	37.000
SWORD OF FARGOAL	C	58.000
ZORK I	D	77.000
ZORK II	D	77.000
ZORK III	D	77.000
SIMULAZIONI		
COMBAT LEADER	D/C	77.000
COMPUTER BASEBALL	D	77.000
PRO TOUR GOLF	D	77.000
RINGSIDE SEAT	D	77.000
NUKEWAR	C	30.000
PLANET MINERS	C	30.000
NORTH ATLANTIC CONVOY	C	30.000
BEACH HEAD	D	43.500

Softgames Vi propone nella versione per il Commodore 64 i programmi sopraelencati. Altre proposte interessanti nell'ambito di vasta scelta di software vi attendono.

I Prezzi si intendono al netto di IVA 18% e franco nostra sede di Torino. Sono soggetti a variazioni senza preavviso per fluttuazioni del \$. Siamo a vostra disposizione presso i nostri uffici. Scriveteci al seguente indirizzo: **SOFTGAMES s.r.l. - 10138 Torino - Via Duchessa Jolanda 9**. Oppure telefonateci allo **(011) 769857** specificando i vostri interessi. La spedizione si effettua contrassegno. Condizioni particolari ai sigg. rivenditori.

occhialuto, in sequito sarà elegante e così via.

Com'è intuitivo, la possibilità di memorizzare una gran quantità di nominativi o di servizi di lunghe frasi esplicative, dipende dalla memoria disponibile. Nel

caso si disponga di un consistente numero di byte (come nel Commodore 64) si può modificare la linea 100.

Quando si desidera smettere verrà posta la domanda se registrare l'archivio oppure no. In ca-

so affermativo si preparerà un nastro cassetta e si seguirà la nota procedura di registrazione. Chi possiede l'unità a dischi non dovrebbe avere difficoltà a modificare la routine di lettura e scrittura dei dati. ■

```
230 GOSUB 630
240 IF B$= "S" THEN I=A%(I): GOTO 210
250 IF B$ <> "N" THEN 230
260 I=A%(I)+1: GOTO 210
270 PRINT "0" RIGHT$(A$(I), LEN(A$(I))-1)
280 PRINT "HO INDOVINATO?" X$
290 GOSUB 630
300 IF B$= "S" THEN 460
310 IF B$ <> "N" THEN 290
320 IF FRE(0) < 100 THEN PRINT "HAI A DISPOSIZIONE ": PRINT FRE(0) "CARATTERI"
330 INPUT "CHE COS'ERA": B$
340 PRINT "CHE DOMANDA FARESTI PARLANDO DI "B$: INPUT C$
350 PRINT "CHE RISPOSTA DARESTI " X$: INPUT D$
360 IF D$ <> "S" THEN 400
370 A$(J) = "@" + B$
380 A$(J+1) = A$(I)
390 GOTO 430
400 IF D$ <> "N" THEN 350
410 A$(J+1) = "@" + B$
420 A$(J) = A$(I)
430 A$(I) = C$
440 A%(I) = J
450 J = J+2: A$(J) = "FINE"
460 PRINT "VUOI GIOCARE ANCORA?"
470 GOSUB 630
480 IF B$ = "S" THEN 190
490 IF B$ <> "N" THEN 470
500 PRINT "DEVO REGISTRARE?"
510 GOSUB 630
520 IF B$ = "N" THEN END
530 IF B$ <> "S" THEN 510
540 OPEN 1,1,1
550 I=0
560 PRINT#1,A$(I)
570 PRINT#1,A%(I)
580 I=I+1
590 IF I <= J THEN 560
600 CLOSE 1
610 PRINT "DIGITA IL PRIMO NOME DELLA SERIE": INPUT B$
620 A$(0) = "@" + B$:J=1: GOTO 190
630 GET B$: IF B$ = "" THEN 630
640 RETURN
```

Un programma assembler

Digitare questo programma non è semplice né molto divertente. Alla fine, però, scoprirete un nuovo modo di programmare.

IL programma assembler inizia alla riga 1111, perchè nelle prime 1111 linee (dalla 0 alla 1110), andrà scritto il programma da assemblare. Il formato di una riga dovrà essere:

numero, virgolette, istruzione in assembler.

Esempio: 10 "lda #234

Vi è la possibilità di inserire dei commenti che dovranno essere preceduti dal punto e virgola (;). Si dovrà dunque scrivere:

10 "lda #234; commento qualunque

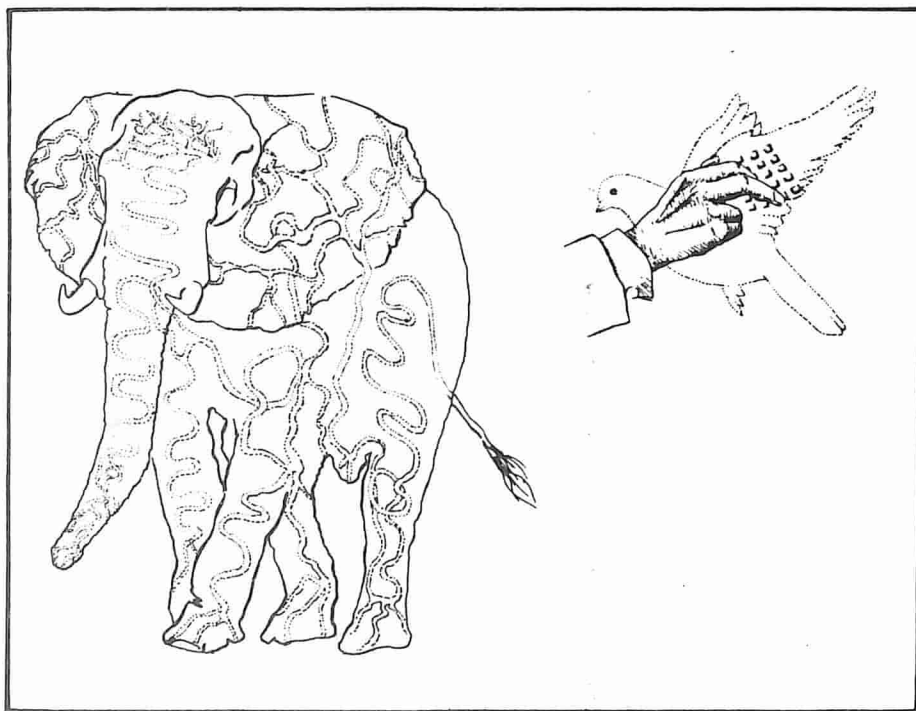
oppure:

10 "; commento qualunque

E' necessario, però, ricordare che la prima istruzione di qualsiasi programma in assembler, dovrà essere del tipo:

10 "*"xxxx

ove al gruppo xxxx deve essere sostituito il valore (nel caso nostro decimale) di partenza di as-



semblaggio. In altri termini (se si vuole che la nostra routine in l.m. cominci dalla locazione esadecimale 033c, bisognerà scrivere:

10 "\$033c ; punto start
il riferimento ad etichette mediante l'uso del prefisso @ (chiocciola), seguito dal nome della label.

Istruzioni del microprocessore 6502							
116 "	SOMMA CON	124 "	ADC #MI	134 "	ADC ,X(MI)	144 "	AND ,(MI)
118 "	RIPORTO	126 "	ADC ,X(MA)	136 "		146 "	AND #MI
120 "	ADC (MA)	128 "	ADC ,Y(MA)	138 "	AND LOGICO	148 "	AND ,X(MA)
122 "	ADC ,(MI)	130 "	ADC /X(MI)	140 "		150 "	AND ,Y(MA)
		132 "	ADC /Y(MI)	142 "	AND (MA)	152 "	AND /X(MI)

154	"	AND /Y(MI)	258	"	SBC ,X(MA)	362	"	BCC UNO	466	"	;
156	"	AND ,X(MI)	260	"	SBC ,Y(MA)	364	"	BCS UNO	468	"	JMP (MA)
158	"	;CONFRONTA	262	"	SBC /X(MI)	366	"	BEQ UNO	470	"	JMP /X(MI)
160	"	;CON	264	"	SBC /Y(MI)	368	"	BMI UNO	472	"	;SALTA ALLA
162	"	ACCUMULATORE	266	"	SBC ,X(MI)	370	"	BNE UNO	474	"	;
164	"	CMP (MA)	268	"	;IMMAGAZZINA	372	"	BPL UNO	476	"	;
166	"	CMP ,(MI)	270	"	;L'	374	"	BVC UNO	478	"	;JSR (MA)
168	"	CMP #MI	272	"	ACCUMULATORE	376	"	BVS UNO	480	"	;
170	"	CMP ,X(MA)	274	"	STA (MA)	378	"	;	482	"	;CARICA X
172	"	CMP ,Y(MA)	276	"	STA ,(MI)	380	"	;IMPLICATI	484	"	;
174	"	CMP /X(MI)	278	"	STA ,X(MA)	382	"	;	486	"	LDX (MA)
176	"	CMP /Y(MI)	280	"	STA ,Y(MA)	384	"	BRK	488	"	LDX ,(MI)
178	"	CMP ,X(MI)	282	"	STA /X(MI)	386	"	CLD	490	"	LDX #MI
180	"	;	284	"	STA /Y(MI)	388	"	CLC	492	"	LDX ,Y(MA)
182	"	OR ESCLUSIVO	286	"	STA ,X(MI)	390	"	CLI	494	"	LDX ,Y(MI)
184	"	;	288	"	;SPOSTAMENTO	392	"	CLV	496	"	;
186	"	EOR (MA)	290	"	;ARITMETICO	394	"	DEX	498	"	;CARICA Y
188	"	EOR ,(MI)	292	"	;A SINISTRA	396	"	DEY	500	"	;
190	"	EOR #MI	294	"	ASL A	398	"	INX	502	"	LDY (MI)
192	"	EOR ,X(MA)	296	"	ASL (MA)	400	"	INY	504	"	LDY ,(MI)
194	"	EOR ,Y(MA)	298	"	ASL ,(MI)	402	"	NOP	506	"	LDY #MI
196	"	EOR /X(MI)	300	"	ASL ,X(MA)	404	"	PHA	508	"	LDY ,X(MA)
198	"	EOR /Y(MI)	302	"	ASL ,X(MI)	406	"	PHP	510	"	LDY ,X(MI)
200	"	EOR ,X(MI)	304	"	;SPOSTAMENTO	408	"	PLA	512	"	;IMMAGAZZINA
202	"	;CARICA L'	306	"	;LOGICO	410	"	PLP	514	"	;
204	"	ACCUMULATORE	308	"	;A DESTRA	412	"	RTI	516	"	;X
206	"	;	310	"	LSR A	414	"	RTS	518	"	STX (MA)
208	"	LDA (MA)	312	"	LSR (MA)	416	"	SEC	520	"	STX ,(MI)
210	"	LDA ,(MI)	314	"	LSR ,(MI)	418	"	SED	522	"	STX ,Y(MI)
212	"	LDA #MI	316	"	LSR ,X(MA)	420	"	SEI	524	"	;IMMAGAZZINA
214	"	LDA ,X(MA)	318	"	LSR ,X(MI)	422	"	TAX	526	"	;
216	"	LDA ,Y(MA)	320	"	;ROTAZIONE	424	"	TAY	528	"	;Y
218	"	LDA /X(MI)	322	"	;A DESTRA	426	"	TSX	530	"	STY (MA)
220	"	LDA /Y(MI)	324	"	;	428	"	TXA	532	"	STY ,(MI)
222	"	LDA ,X(MI)	326	"	ROR A	430	"	TXS	534	"	STY ,X(MI)
224	"	;	328	"	ROR (MA)	432	"	TYA	536	"	;VERIFICA
226	"	;OR LOGICO	330	"	ROR ,(MI)	434	"	;DECREMENTA	538	"	;
228	"	;	332	"	ROR ,X(MA)	436	"	;LA MEMORIA	540	"	;DI BIT
230	"	ORA (MA)	334	"	ROR ,X(MI)	438	"	;	542	"	BIT (MA)
232	"	ORA ,(MI)	336	"	;ROTAZIONE	440	"	DEC (MA)	544	"	BIT ,(MI)
234	"	ORA #MI	338	"	;A SINISTRA	442	"	DEC ,(MI)	546	"	;CONFRONTA
236	"	ORA ,X(MA)	340	"	;	444	"	DEC ,X(MA)	548	"	;
238	"	ORA ,Y(MA)	342	"	ROL A	446	"	DEC ,X(MI)	550	"	;CON X
240	"	ORA /X(MI)	344	"	ROL (MA)	448	"	;INCREMENTA	552	"	CPX (MA)
242	"	ORA /Y(MI)	346	"	ROL ,(MI)	450	"	;LA MEMORIA	554	"	CPX ,(MI)
244	"	ORA ,X(MI)	348	"	ROL ,X(MA)	452	"	;	556	"	CPX #MI
246	"	;SOTTRAE CON	350	"	ROL ,X(MI)	454	"	INC (MA)	558	"	;CONFRONTA
248	"	;	352	"	;	456	"	INC ,(MI)	560	"	;CON Y
250	"	;RIPORTO	354	"	@UNO	458	"	INC ,X(MA)	562	"	;
252	"	SBC (MA)	356	"	;	460	"	INC ,X(MI)	564	"	CPY (MA)
254	"	SBC ,(MI)	358	"	;DIRAMAZIONI	462	"	;	566	"	CPY ,(MI)
256	"	SBC #MI	360	"	;	464	"	;SALTA A	568	"	CPY #MI

Il listato del programma

```
1111 REM"
1112 REM"
1113 REM"    DI VIZIO    MAURIZIO
1114 REM"
1115 REM"    TEL 039 / 743092 .
1116 REM"
1117 REM"
1118 REM"    -- ASSEMBLATORE 6502 --
1119 REM"
5000 CLOSE3:GOTO6840
5001 CLR:INPUT"VOSTRI USANDO IL PET ";SU$:IFLEFT$(SU$,1)="S"THENWU=41:GOTO5010
5002 WU=44
5010 DIML$(8),VL(8):PRINT" "
5020 IB=PEEK(WU)*256+PEEK(WU-1)-1:YY=IB
5030 PRINT" "
5040 GOTO5090
5050 :
5060 S$=""
5070 IFPEEK(IB)=0THENRETURN
5080 S$=S$+CHR$(PEEK(IB)):IB=IB+1:GOTO5070
5090 :
5100 PRINT"LINEA/PROG.COUNTER:";PRINT
5110 VOLTA=1
5120 LA=PEEK(IB+4)*256+PEEK(IB+3):IFLA=1111THEN5490
5130 PRINTLA,PC:IB=IB+5
5140 GOSUB5050
5150 M$=CHR$(34)+"*":IFV0=1ANDLEFT$(S$,2)<>M$THENPRINT"DA DOVE ALLOCO ?","ERR.
IN";LA:END
5160 IFLEFT$(S$,2)=M$THEN 6750
5170 IFLEFT$(S$,1)<>CHR$(34)THENPRINT"ERRORE IN";LA:END
5180 IF MID$(S$,2,1)=";"THEN5120
5190 IF MID$(S$,2,1)="@"THEN5410
5200 IFMID$(S$,2,1)="="THEN6500
5210 GOTO5300
5220 FORK=1TOLEN(S$):IFMID$(S$,K,1)="/"THEN5240
5230 NEXT:PRINT"ERRORE IN";LA:END
5240 S$=MID$(S$,K+1,LEN(S$))
5250 FORK=1TOLEN(S$):IFMID$(S$,K,1)="/"THEN5280
5260 IFK>80THENPRINT"TESTO TROP. LUNGO IN";LA:END
5270 NEXT:PRINT"ERRORE IN";LA:END
5280 S$=MID$(S$,1,K-1):PC=PC+LEN(S$):IFPR#0THEN5120
5290 PR=0:FORK=1TOLEN(S$):H=ASC(MID$(S$,K,1)):GOSUB6630:PC=PC+1:NEXT:GOTO5570
5300 FORK=1TOLEN(S$):IFMID$(S$,K,1)=" "ORMID$(S$,K,1)=CHR$(34)THENNEXTK
5310 S$=MID$(S$,4+K,LEN(S$)-K+1)
5320 IF LEFT$(S$,1)=" "ORLEFT$(S$,1)=";"ORLEFT$(S$,1)=""THENPC=PC+1:GOTO5120
5330 R$=MID$(S$,2,1):IFLEFT$(S$,1)="A"AND(R$=" "ORR$=";"ORR$="")THENPC=PC+1:GOT
05120
5331 SA$=LEFT$(S$,1)
5340 IFSA$="#"ORSAS$="$"THENPC=PC+2:GOTO5120
5350 IFSA$="<"THENPC=PC+3:GOTO5120
5360 IFSA$=","THENPC=PC+3:GOTO5120
```



```

5370 IFSA$="." THEN PC=PC+2:GOTO5120
5380 IFSA$="=" THEN IFMIS$(S$,2,1)="(" THEN PC=PC+3:GOTO5120
5390 IFSA$=">" THEN PC=PC+2:GOTO5120
5400 PC=PC+2:GOTO5120
5410 :
5420 S$=RIGHT$(S$,LEN(S$)-2):FOR K=1 TO LEN(S$):G$=MID$(S$,K,1):IFG$=" " ORG$=";" TH
EN5440
5430 NEXT:GOTO5450
5440 S$=MID$(S$,1,K-1)
5450 FOR N=0 TO I:IFL$(N)=S$ THEN PRINT "LAB. RIPET. ERRORE IN";LA:END
5460 NEXT:I=I+1
5470 L$(I)=S$:VL(I)=PC
5480 GOTO5120:
5490 :
5500 PRINT "S=====":PRINT
5510 PRINT "ETICHETTA/", "VALORE":PRINT
5520 PRINT:FOR N=1 TO I:PRINT L$(N), "VL(N):NEXT:PRINT
5530 PRINT "S=====
5540 FOR BL=1 TO 2500:NEXT:PRINT "0000"
5550 PRINT "*** CODICE OGGETTO ***"
5560 FF=1:IB=YY:PC=0:TP=PC-1
5570 S$="":LA=PEEK(1B+4)*256+PEEK(1B+3):IFLA=1111 THEN 6830
5580 1B=1B+5
5590 GOSUB5050
5600 IF LEFT$(S$,1)<>CHR$(34) THEN PRINT "ERRORE IN";LA:END
5610 IF MID$(S$,2,1)=";" THEN 5570
5620 IF MID$(S$,2,1)="=" THEN 5570
5630 IF MID$(S$,2,1)="@" THEN 5570
5640 IF MID$(S$,2,1)="*" THEN:IN=1:GOTO5150
5650 S$=MID$(S$,2,LEN(S$)):FOR K=1 TO LEN(S$):IF MID$(S$,K,1)=" " OR MID$(S$,K,1)=":"
THEN NEXT K
5660 SA$=MID$(S$,K,3)
5670 SP$=MID$(S$,K+4,LEN(S$))
5680 FOR K=1 TO LEN(SP$):IF MID$(SP$,K,1)=";" OR MID$(SP$,K,1)=" " THEN K=LEN(SP$):GOTO
5700
5690 NEXT:GOTO5710
5700 SP$=MID$(SP$,1,K-1)
5710 FOR K=1 TO LEN(SP$):G$=MID$(SP$,K,1)
5720 IFG$=" " ORG$=";" ORG$=";" THEN 5740
5730 NEXT:GOTO5750
5740 SP$=MID$(SP$,1,K)
5750 :
5760 ST$=LEFT$(SP$,1)
5770 IF ST$="ORST$;" THEN PC=PC+1:BB$="IMP":GOTO5900
5780 R$=MID$(SP$,2,1):IF ST$="A" AND (R$=" " OR R$=";" OR R$=">") THEN PC=PC+1:BB$="010":
TS=1:GOTO5900
5790 IF ST$="#" OR ST$="$" THEN PC=PC+2:BB$="010":GOTO5900
5800 IF ST$="," AND LEFT$(SP$,2)=",X" THEN PC=PC+3:BB$="111":GOTO5900
5810 IF ST$="," AND LEFT$(SP$,2)=",Y" THEN PC=PC+3:BB$="110":GOTO5900
5820 IF ST$="(" THEN PC=PC+3:BB$="011":GOTO5900
5830 IF ST$="." AND LEFT$(SP$,2)=",X" THEN PC=PC+2:BB$="101":GOTO5900
5840 IF ST$="." AND LEFT$(SP$,2)=",Y" THEN PC=PC+2:BB$="10":GOTO5900
5850 IF ST$="." THEN PC=PC+2:BB$="001":GOTO5900
5860 IF ST$="/" AND LEFT$(SP$,2)="/X" THEN PC=PC+2:BB$="000":GOTO5900
5870 IF ST$="/" AND LEFT$(SP$,2)="/Y" THEN PC=PC+2:BB$="100":GOTO5900
5880 IF ST$="/" THEN PC=PC+3:BB$="1":GOTO5900
5890 BB$="REL":PC=PC+2:GOTO5900
5900 :
5910 IF SA$="LDY" AND BB$="10" THEN:K:BB$="101":GOTO5940

```



```

5920 IFSA$="LDX"ANDBB$="110"THENBB$="111":GOTO5940
5930 IFSA$="LDX"ANDBB$="10"THENBB$="110":FJ=1:GOTO5940
5940 READ C1$,C2$:IFC1$="-1"THEN5970
5950 IF C1$=SA$THEN5990
5960 GOTO5940
5970 :
5980 PRINT"ERRORE IN";LA:END
5990 RESTORE
6000 M$=MID$(SA$,3,1):IFM$="X"ORM$="Y"THEN:IFBB$="010"THENBB$="000"
6010 VL$="":P=1:FORK=1T08:IFMID$(C2$,K,1)="B"THENVL$=VL$+MID$(BB$,P,1):P=P+1:GO
T06032
6020 VL$=VL$+MID$(C2$,K,1)
6030 NEXT:V=0:FORK=0T07:IFVAL(MID$(VL$,K+1,1))=1THENV=V+2*(7-K)
6040 NEXT:H=V:GOSUB6630
6050 IFBB$="IMP"OR(BB$="010"ANDTS=1)THENV=0:GOTO5570
6060 VM=1
6070 IFBB$="101"ORBB$="10"ORBB$="001"ORBB$="000"ORBB$="100"THENVM=0
6080 IFFJ=1ANDBB$="110"THENFJ=0:VM=0:REM 0,Y DI LDX
6090 IFBB$="REL"THEN6200
6100 M$=MID$(SA$,3,1):IFM$="X"ORM$="Y"THEN:IFBB$="000"THENBB$="010"
6110 IFBB$="010"THEN::GOTO6290
6120 GOSUB6420
6130 IF LEFT$(SP$,1)="#" OR LEFT$(SP$,1)="$"THEN6690
6140 FORN=1TOI:IFL$(N)=SP$THEN:GOTO6160
6150 NEXT:PRINT"ERRORE IN";LA:END
6160 IFVL(N)>255ANDVM=0THENPRINT"ERRORE IN";LA:END
6170 IFVL(N)=<255ANDVM=0THENH=VL(N):GOSUB6630:GOTO5570
6180 IFVL(N)=<255ANDVM=1THENH=VL(N):GOSUB6630:H=0:GOSUB6630:GOTO5570
6190 IFVL(N)>255THENH=VL(N)-256*INT(VL(N)/256):GOSUB6630:H=INT(VL(N)/256):GOSUB
6630:GOTO5570
6200 :
6210 FORK=1TOLEN(SP$):M$=MID$(SP$,K,1):IFM$=" "ORM$=";"THEN6230
6220 NEXT:GOTO6240
6230 SP$=MID$(SP$,1,K-1)
6240 FORK=1TOI:IFL$(K)=SP$THEN6260
6250 NEXT:PRINT"ERRORE IN";LA:END
6260 D=ABS(PC-VL(K)):IFD>127THENPRINT"ERR. RANGE DEL RELATIVO IN";LA:END
6270 IFPC>VL(K)THENH=256-(PC-VL(K)):GOSUB6630:GOTO5570
6280 H=VL(K)-PC:GOSUB6630:GOTO5570
6290 :
6300 X$=MID$(SP$,2,LEN(SP$)):IFASC(LEFT$(X$,1))<48ORASC(LEFT$(X$,1))>57THEN6330
6310 IF LEFT$(SP$,1)="#" THEN6340
6320 X=VAL(X$):H=X:GOSUB6630:GOTO5570
6330 :
6340 IFLEFT$(SP$,1)="#" THEN7270
6350 FORK=1TOLEN(X$):IFMID$(X$,K,1)="#"ORMID$(X$,K,1)="#" THEN6370
6360 NEXT:GOTO6380
6370 X$=MID$(X$,1,K-1)
6380 FORK=1TOI:IFL$(K)=X$THENH=VL(K):GOTO6400
6390 NEXT:PRINT"ERRORE IN";LA:END
6400 IFH>255THENPRINT"ERR. RANGE IN ";LA:END
6410 GOSUB6630:GOTO5570
6420 :
6430 FORK=1TOLEN(SP$):IFMID$(SP$,K,1)="#" THEN6450
6440 NEXT:PRINT"ERRORE IN";LA:END
6450 SP$=MID$(SP$,K+1,LEN(SP$))
6460 FORK=1TOLEN(SP$):IFMID$(SP$,K,1)="#" THEN6480
6470 NEXT:PRINT"ERRORE IN";LA:END
6480 SP$=MID$(SP$,1,K-1)

```




```

6490 :RETURN
6500 :
6510 S#=MID$(S$,3,LEN(S$)):FORK=1TOLEN(S$):IFMID$(S$,K,1)=" "THEN6530
6520 NEXT:PRINT"ERRORE IN";LA:END
6530 W1#=MID$(S$,1,K-1)
6540 W2#=MID$(S$,1+K,LEN(S$))
6550 IFLEFT$(W2$,1)="#"THEN7220
6560 IF LEFT$(W2$,1)="#"THEN6580
6570 PRINT"ERRORE IN";LA:END
6580 W2#=MID$(W2$,2,LEN(W2$)):FORK=1TOI
6590 IFW1$=L$(K)THENPRINT"ERRORE IN";LA:END
6600 NEXT
6610 I=I+1:VL(I)=VAL(W2$):L$(I)=W1$
6620 GOTO5120
6630 :
6640 :
6650 POKEJ+TP,H
6660 TP=TP+1:H$=STR$(H)
6670 PRINTH$,
6680 RETURN
6690 :
6700 IFLEFT$(SP$,1)="#"THENA$=MID$(SP$,2,4):GOSUB7110:H=V:M=1:GOTO6730
6710 M=ASC(MID$(SP$,2,3)):IFM<48ORM>57THENPRINT"ERRORE IN";LA:END
6720 H=VAL(MID$(SP$,2,LEN(SP$))):VM=1
6730 IFBB$="101"ORBB$="10"ORBB$="001"ORBB$="000"ORBB$="100"THENVM=0
6740 N=0:VL(0)=H:GOTO6160
6750 :
6760 S#=MID$(S$,3,LEN(S$))
6770 IFLEFT$(S$,1)="#"THEN7070
6780 IF LEFT$(S$,1)="#"THENPRINT"ERRORE IN";LA:END
6790 S#=MID$(S$,2,LEN(S$))
6800 PC=VAL(S$):J=PC:TP=0:IF FF=1THENPRINT:PRINT:RR=0:H$="===":GOSUB6670
6810 VOLTE=2:IF IN=1THEN IN=0:GOTO5570
6820 GOTO5120
6830 PRINT:PRINT"FINE ASSEMBLAGGIO.";FORK=0TO3500:NEXT:POKE198,0
6840 :
6850 PRINT" ";CHR$(14);CHR$(8)
6851 PRINT"===== "
6852 PRINT" ASSEMBLATORE "
6853 PRINT"===== "
6860 PRINT:PRINT"1) ASSEMBLAGGIO":PRINT"2) SALVATAGGIO"
6870 PRINT"3) CARICAMENTO"
6880 PRINT"4) LISTARE SU VIDEO"
6902 PRINT"5) FINE PROGRAMMA ."
6903 PRINT"QUALE OPZIONE ?";
6910 GETA$:IFVAL(A$)<1ORVAL(A$)>5THEN6910
6911 PRINT" "
6930 IFA$="4"THENPRINT" ":LIST-1110:END
6940 IFA$="1"THEN5001
6950 IFA$="5"THENPRINT" ":END
6951 PRINT"STAI USANDO ":PRINT"IL PET (S/N)":INPUTOC$:IFLEFT$(OC$,1)="N"THEN69
60
6952 PRINT"LEGGERE L' ARTICOLO"
6953 FORYB=1TO6000:NEXT:RUN1111
6960 RESTORE:PRINT" "
6970 READA$,B$:IFA$<"-1"THEN6970
6980 INPUT"PERIFERICA (1/8)":P:INPUT"START (DEC)":V:INPUT"END (DEC)":W
6990 INPUT"NAMEFILE":V$:PRINT:PRINT:R=540:FORJ=1TO43:READT:POKER+J+5,T:NEXT:PO
KE549,P

```

```

7000 IFP=8THENCLOSE1:IF W THEN OPEN1,8,15,"S0:"+V$
7010 T=LEN(V$):POKE25,T:U=1:S=256*PEEK(50)+PEEK(49)+T
7020 FORJ=1TOT:POKES-J,ASC(RIGHT$(V$,J)):NEXT:SYS546:U=V:T=252:GOSUB7060:IFW=0T
HEN7040
7030 U=W:T=254:GOSUB7060:SYS579:RUN5000
7040 FORJ=0T05:POKER+J,PEEK(45+J):NEXT:SYS569:FORJ=0T05:POKE45+J,PEEK(R+J):NEXT
7050 RUN5000
7060 POKET,INT(U/256):POKET-1,U-256*PEEK(T):RETURN
7070 S$=MID$(S$,2,LEN(S$)):FORK=1TOLEN(S$):G$=MID$(S$,K,1):IFG$=" "ORG$=":"THEN
7090
7080 NEXT
7090 IFK<>3ANDK<>5THENPRINT"ERRORE IN";LA:END
7100 A$=MID$(S$,1,K-1):GOSUB7110:S$=STR$(V):GOTO6800
7110 :
7120 A$=MID$(A$,1,4):IFLEN(A$)<4THENGOSUB7150:V=A:RETURN
7130 W$=A$:A$=MID$(W$,1,2):GOSUB7150:W=A
7140 A$=MID$(W$,3,2):GOSUB7150:V=256*W+A:RETURN
7150 A1$=LEFT$(A$,1):A2$=RIGHT$(A$,1)
7160 IF A1$<"A" THEN7180
7170 A1$=STR$(ASC(A1$)-55)
7180 IF A2$<"A" THEN7200
7190 A2$=STR$(ASC(A2$)-55)
7200 A=VAL(A1$)*16+VAL(A2$)
7210 RETURN
7220 W2$=MID$(W2$,2,LEN(W2$)):FORK=1TOLEN(W2$):G$=MID$(W2$,K,1)
7230 IFG$=" "ORG$=":"THEN7250
7240 NEXT
7250 IFK<>3ANDK<>5THENPRINT"ERRORE IN";LA:END
7260 A$=MID$(W2$,1,K-1):GOSUB7110:W2$="#" +STR$(V):GOTO6580
7270 FORK=1TOLEN(X$):G$=MID$(X$,K,1)
7280 IFG$=" "ORG$=":"THEN7300
7290 NEXT
7300 IFK<>3ANDK<>5THENPRINT"ERRORE IN";LA:END
7310 A$=MID$(X$,1,K-1):GOSUB7110:H=V:GOTO 6400
7320 DATA DC,011BBB01,AND,001BBB01,ASL,000BBB10,BCC,10010000,BCS,10110000,BEQ,1
1110000
7330 DATABIT,001BBB00,BMI,00110000,BNE,11010000,BPL,00010000,BRK,00000000,BVC,0
1010000
7340 DATABYS,01110000,CLC,00011000,CLD,11011000,CLI,01011000,CLV,10111000,CMF,1
10BBB01
7350 DATACPX,111BBB00,CPY,110BBB00,DEC,110BB110,DEX,11001010,DEY,10001000,EOR,0
10BBB01
7360 DATA INC,111BB110,INX,11101000,JMP,01B01100,JSR,00100000,LDA,101BBB01
7370 DATALDX,101BBB10,LDY,101BBB00,LSR,01BBB110,NOP,11101010,ORA,000BBB01,PHA,0
1001000
7380 DATAPHP,00001000,PLA,01101000,PLP,00101000,ROL,001BBB10,ROR,011BBB10,RTI,0
1000000
7390 DATARTS,01100000,SBC,111BBB01,SEC,00111000,SED,11111000,SEI,01111000,STA,1
00BBB01
7400 DATA STX,100BB110,STY,100BB100,TAX,10101010,TAY,10101000,TSX,10111010,TXA,1
0001010
7410 DATATXS,10011010,TYA,10011000,INY,110010000,-1,-1
7420 DATA 169,1,162,1,160,0,32,186,255,165,25,166,49,164,50,32,189,255,169,000
7421 DATA 133,157
7430 DATA 96,169,0,166,251,164,252,32,213,255,96,169,251,166,253,164,254,32,216,
255,96
7440 REM FORZA KEY
7450 PRINT#3:CLOSE3: RUN1111
READY.

```

Programmi di prova per il Vic 20

Un bip
ad ogni return

```

10 "##033C;
11 ";
12 "; INIZIALIZZAZIONE
13 ";
14 " LDA (#772);
15 " STA (#1000);
16 " LDA (#773);
17 " STA (#1001);
18 " LDA #132;
19 " STA (#772);
29 " LDA #3;
30 " STA (#773);
31 " RTS ;
32 ";
33 "; GENERA SONORO
34 ";
35 "##900;
36 "=VOL=#36878;
37 "=SON=#36875;
38 "=NOTA=#211;
39 " STA .(#1);
40 " STX .(#2);
41 " STY .(#3);
42 " LDA #15;
43 " STA (VOL);
44 " LDA #NOTA;
45 " STA (SON);
46 " LDY #255
47 "@UNO;
48 " LDX #255
49 "@DUE;
50 " DEX ;
51 " BNE DUE;
52 " DEY ;
53 " BNE UNO;
54 " LDA #0;
55 " STA (VOL);
56 " LDA .(#1);
57 " LDX .(#2);
58 " LDY .(#3);
59 " JMP /(#1000);
60 ";
61 ";

```

Per lo scroll
del video

```

10 "##828
11 ";
12 ";
14 " LDX #230
15 "@UNO
16 " LDA ,X(#4349)
17 " STA ,X(#4371)
18 " LDA ,X(#38141)
19 " STA ,X(#38163)
20 " DEX
21 " CPX #255
22 " BNE UNO
23 ";
24 " LDX #252
25 "@DUE

```

```

26 " LDA ,X(#4096)
27 " STA ,X(#4118)
28 " LDA ,X(#37888)
29 " STA ,X(#37910)
30 " DEX
31 " CPX #255
32 " BNE DUE
33 ";
34 " LDX #22
35 " LDA #32
36 "@TRE
37 " STA ,X(#4095)
38 " DEX
39 " BNE TRE
40 " RTS
41 ";
42 ";

```

Per rinumerare

```

100 "##828
102 " LDA #10
104 " STA .(#4)
105 " LDA #0
106 " STA .(#5)
107 " LDA #1
108 " STA .(#0)
109 " LDA #18
110 " STA .(#1)
111 "@DUE
112 " LDY #0
113 " LDA /Y(#0)
114 " BNE UNO
115 " RTS
116 "@UNO
117 " STA .(#6)
118 " INY
119 " LDA /Y(#0)
120 " STA .(#7)
121 " LDA .(#4)

```

```

122 " INY
123 " STA /Y(#0)
124 " LDA .(#5)
125 " INY
126 " STA /Y(#0)
127 " CLC
128 " LDA .(#4)
129 " ADC #10
130 " STA .(#4)
131 " BCC TRE
132 " LDA .(#5)
133 " ADC #0
134 " STA .(#5)
135 "@TRE
136 " LDA .(#6)
137 " STA .(#0)
138 " LDA .(#7)
139 " STA .(#1)
140 " LDA #0
141 " BEQ DUE
142 ";

```

Organetto

```

10 "##033C
11 ";
12 "=ON=$0F
13 "=OFF=$00
14 "=KEY=$CB
15 "=F7=$BF
16 "=NT=$40
17 ";
18 " LDA #ON
19 " STA ($900E)

```

```

20 "@UNO
21 " LDA .(KEY)
22 " CMP #NT
23 " BEQ UNO
24 " ADC #80
25 " STA ($900B)
26 " CMP #F7
27 " BNE UNO
28 " LDA #OFF
29 " STA .($C6)
30 " STA ($900E)
31 " RTS

```

COMMODORE



Se stai comprando un personal computer prova a farti queste domande:

1. Chi è oggi il più affidabile?
2. Chi dà la possibilità di scegliere fra più sistemi?
3. Chi fornisce soluzioni, subito,

in una gamma vastissima?

4. Chi propone il miglior rapporto fra costi e prestazioni?
5. Chi ti dà una così grande esperienza ed assistenza?

A tutte le domande puoi rispondere con

ORE, IL N°1



un solo nome: Commodore Computer.

Anche per questo Commodore
è il Numero 1. In Europa e in Italia.

Sei in buone mani.

Commodore Italiana Spa
Milano, telefono 02/6125651

 **commodore**
COMPUTER

10 "@routine

Il precedente esempio definisce l'etichetta "routine" utile per esempio, nel caso si voglia scrivere un relativo diretto ad un indirizzo variabile:

es: 10 "bne routine

Il numero massimo di label disponibili è scritto nel doppio dimensionamento di vettori di riga 5010. Nel caso si vogliono più di 8 etichette, per esempio, in numero x, si dovrà digitare:

```
5010 clr:dim l$(x), vl(x): print  
"  " "
```

Si ricordi, inoltre, che nello scrivere una linea, tra l'istruzione mnemonica (es: lda) e il relativo argomento (es: #234), occorre inserire uno spazio.

La capacità di riferirsi a valori decimali (#) oppure esa (\$), si aggiunge alla possibilità di definire delle costanti, di più lettere e in qualsiasi punto del programma. Per esempio, volendo assegnare a k il valore decimale 1234, si scriverà:

```
10 "=k=#1234, eventuale  
commento.
```

Per utilizzare tale costante, per esempio, in modo assoluto, si deve scrivere:

```
10 " lda (k) di cento più vantag-  
gioso di:
```

```
10 " lda (#1234)
```

Chiarito come va scritto il programma da assemblare, vediamo come gira il vero e proprio assembler.

Come detto precedentemente, occorre digitare "run 1111"; in seguito tale ordine il video mostra una serie di opzioni, ottenibili con la pressione dei tasti-funzio-

ne, 1,3,5,7, ed 8, come, appunto, specificato nel menu. Vediamole singolarmente.

Assemblaggio

Premendo (f1) l'assemblatore effettua una prima verifica, riga per riga, in modo da controllare che tutte le labels siano corrette e che siano, altresì, esatti gli assegnamenti delle costanti. Il video mostra due colonne di valori: la prima indica il numero della linea che sta per essere esaminata, la seconda è la successione delle locazioni di memoria utilizzate dall'assemblatore per allocare il programma; in seguito lo schermo mostra tutte le labels e costanti, con il relativo valore espresso in decimale.

Premendo la barra spaziatrice, l'assemblatore provvede ad un ulteriore test e solo se tutto è corretto comincia a tradurre l'assembler in Linguaggio Macchina (L.M.).

Nel caso si riscontrino errori, verrà segnalato il numero della riga non corretta, col conseguente arresto della fase di traduzione.

In tal caso, bisogna verificare che la sintassi di tale riga sia esatta, se in essa siano state usate costanti non nominate precedentemente, se il salto del relativo sia troppo esteso, se il relativo chiami una label innominata, e così via,

Eseguita la correzione necessaria, riprendere nuovamente la traduzione, mediante "run 1111"

Corretti eventuali errori, termina-

to l'assemblaggio vero e proprio, il programma assembler mostra sullo schermo la primitiva videata, che avverte delle diverse opzioni.

I tasti (f3) o (f5), provvedono alla fase di lettura/scrittura su nastro/disco del programma precedentemente assemblato.

Facciamo l'ipotesi che la nostra routine in l.m. sia allocata da 828 (dec) sino a 1000 (dec). Volendo registrare su nastro o disco tale zona, scelta la periferica (nastro=1 / disco=8) bisognerà rispondere agli input nel modo seguente:

```
start? 828
```

```
end? 1000
```

Per ricaricare invece da periferica il programma in l.m., precedentemente registrato, si risponderà:

```
start? 828
```

```
end ? 0
```

Listati

Digitando (f7) si otterrà sul video l'intero listato del programma da assemblare.

Desiderando invece il list su stampante, si premerà (f8).

Uscita

Alla pressione del tasto-funzione n.5, si esce dal programma assembler, cosa utile (e più elegante che premere restore + run/stop), nel caso si vogliano apportare modifiche o aggiunte al list del programma da assemblare.

**IMPARA
A PROGRAMMARE
CON IL VIC**



*** DISPENSA N. 7 ***

Scrivi anche tu

La collaborazione a Commodore Computer Club è aperta a tutti gli utenti di sistemi CBM che abbiano sviluppato programmi e routine originali e d'interesse generale o che desiderino fare partecipi gli altri lettori delle scoperte e delle osservazioni fatte per una migliore utilizzazione di queste macchine.

Allegate un breve articolo in cui spiegate a che serve il programma, come funziona, su quale computer gira e con quale configurazione (espansa, non espansa, ecc.).

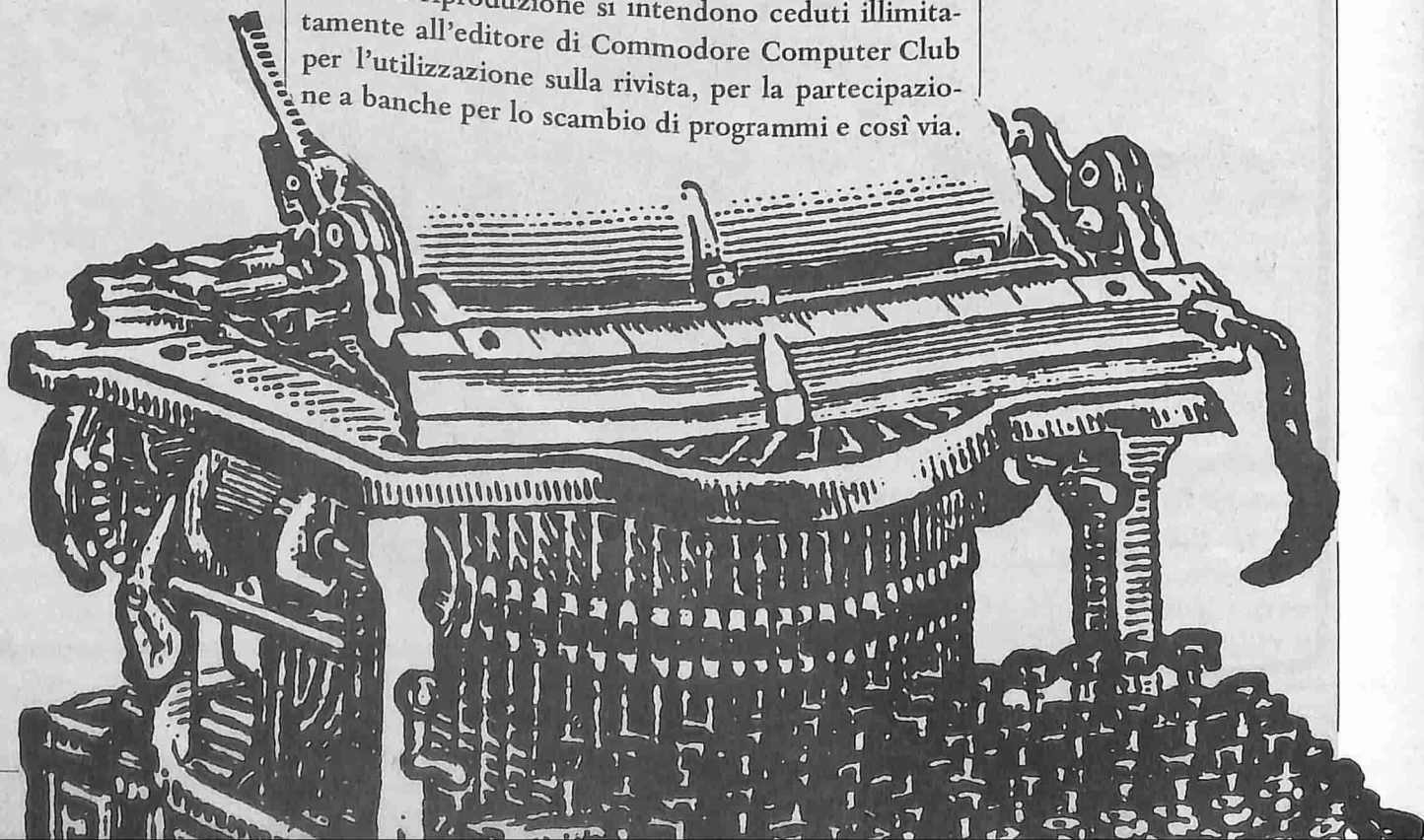
- Iniziate tutti i listati con delle Rem indicanti il titolo del programma, il nome, l'indirizzo ed il telefono dell'autore.

- Inviare i listati su disco o su cassetta, (su carta verranno irrimediabilmente cestinati).

Per compensarvi per questa spesa la redazione vi spedisce a stretto giro di posta due cassette e/o dischetti nuovi per ciascuna cassetta da voi inviata, indipendentemente dal fatto che il programma registrato venga accettato per la pubblicazione.

Ciascun articolo e/o programma pubblicato, invece, verrà compensato con l'invio d'una intera scatola di cassette e/o dischetti.

Gli autori, naturalmente, si assumono ogni responsabilità circa l'originalità dei lavori inviati i cui diritti di riproduzione si intendono ceduti illimitatamente all'editore di Commodore Computer Club per l'utilizzazione sulla rivista, per la partecipazione a banche per lo scambio di programmi e così via.



Tastiera

Un programma musicale che potrebbe interessare gli utenti del Vic

LE capacità musicali del Vic possono essere sfruttate in moltissimi programmi; in generale, però, la difficoltà principale che si incontra consiste nell'associare i tasti del computer alle note da suonare.

Ecco un programma che visualizza sullo schermo una tastiera con 15 note (dal do al mi dell'ottava successiva) con possibilità d'uso delle tre tonalità, mediante i tasti F1, F3, F5.

Premendo i tasti relativi alle note si vedrà contemporaneamente una pallina spostarsi, indicando la nota che viene suonata in quel momento.

Con un po' di pratica sarà quindi possibile suonare guardando solo la tastiera sullo schermo.

Spiegazioni sul listato:

- le linee 10-60 unitamente al sottoprogramma 2000 e al blocco dati 3000-3004 fanno apparire sullo schermo la tastiera;
- le linee 74-108 richiamano i sottoprogrammi 1000-1170, che

generano i suoni e fanno spostare la pallina;

- le linee 110-114 richiamano i sottoprogrammi 1180-1200, che cambiano la tonalità.

Se le note non dovessero cor-

rispondere ad altri strumenti da voi in possesso, agire sulle linee 1000, 1010, 1020, ..., 1170. ■

Giampaolo Cervone

Via Vincenzo Riolo, 10 - 90139 Palermo
Tel. 580061

```

1 REM ***          TASTIERA MUSICALE          ***
2 REM *** PER VIC 20 SENZA ESPANSIONI ***
3 REM *** GIANPAOLO CERVONE -PALERMO- ***
4 REM *** TEL. 580061 VIA V.RIOLO N.10***
5 POKE36879,93:PRINT"J"
10 CA=7680:CO=38400
11 V=36878:S=36876
12 J=99
15 FORK=286T0306
20 GOSUB2000:NEXT
22 J=93
25 FORK=196T0284STEP2
30 GOSUB2000:NEXT
35 READK:IFK=-190T047
40 GOSUB2000
45 GOTO35
47 J=160
50 READK:IFK=-100T067
55 GOSUB2000
60 GOTO50
67 PRINT"### M E T Y U O P"
68 PRINT" A S D F G H J K L : "
69 POKEY,15
70 PRINT"#####F1=ALTI":PRINT
71 PRINT"F3=MEDI":PRINT
72 PRINT"F5=BASSI":PRINT

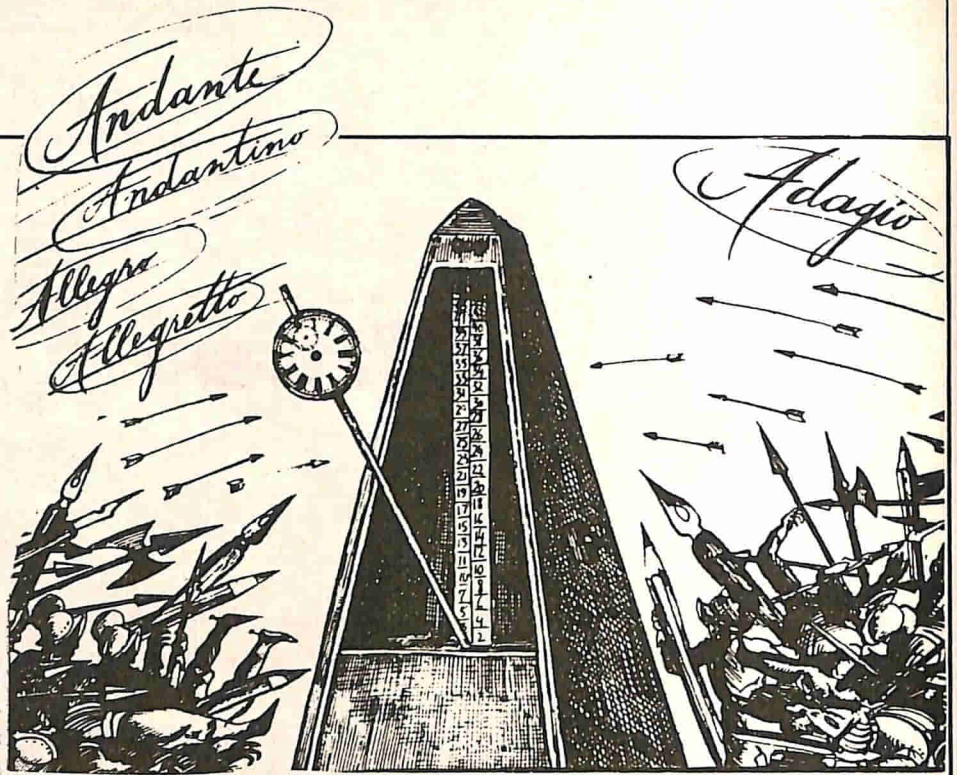
```



```

73 PRINT"SPAZIO=STOP"
74 GETA$
75 IFA$="A"THEN GOSUB1000
78 IFA$="W"THEN GOSUB1010
80 IFA$="S"THEN GOSUB1020
82 IFA$="E"THEN GOSUB1030
83 IFA$="D"THEN GOSUB1040
84 IFA$="F"THEN GOSUB1050
86 IFA$="T"THEN GOSUB1060
88 IFA$="G"THEN GOSUB1070
90 IFA$="Y"THEN GOSUB1080
92 IFA$="H"THEN GOSUB1090
94 IFA$="U"THEN GOSUB1100
96 IFA$="J"THEN GOSUB1110
98 IFA$="K"THEN GOSUB1120
100 IFA$="O"THEN GOSUB1130
102 IFA$="L"THEN GOSUB1140
104 IFA$="P"THEN GOSUB1150
106 IFA$=":"THEN GOSUB1160
108 IFA$=" "THEN GOSUB1170
110 IFA$=CHR$(133)THEN GOSUB1180
112 IFA$=CHR$(134)THEN GOSUB1190
114 IFA$=CHR$(135)THEN GOSUB1200
120 GOTO74
1000 POKES,189
1001 GOSUB2000
1002 POKECA+265,81:POKECO+265,0:K=265:J=32
1005 RETURN
1010 POKES,193
1011 GOSUB2000
1012 POKECA+156,81:POKECO+156,0:K=156:J=160
1015 RETURN
1020 POKES,196
1021 GOSUB2000
1022 POKECA+267,81:POKECO+267,0:K=267:J=32
1025 RETURN
1030 POKES,199
1031 GOSUB2000
1032 POKECA+158,81:POKECO+158,0:K=158:J=160
1035 RETURN
1040 POKES,203
1041 GOSUB2000
1042 POKECA+269,81:POKECO+269,0:K=269:J=32
1045 RETURN
1050 POKES,205
1051 GOSUB2000
1052 POKECA+271,81:POKECO+271,0:K=271:J=32
1055 RETURN
1060 POKES,208
1061 GOSUB2000
1062 POKECA+162,81:POKECO+162,0:K=162:J=160
1065 RETURN

```



```

1070 POKES,211
1071 GOSUB2000
1072 POKECA+273,81:POKECO+273,0:K=273:J=32
1075 RETURN
1080 POKES,213
1081 GOSUB2000
1082 POKECA+164,81:POKECO+164,0:K=164:J=160
1085 RETURN
1090 POKES,216
1091 GOSUB2000
1092 POKECA+275,81:POKECO+275,0:K=275:J=32
1095 RETURN
1100 POKES,218
1101 GOSUB2000
1102 POKECA+166,81:POKECO+166,0:K=166:J=160
1105 RETURN
1110 POKES,220
1111 GOSUB2000
1112 POKECA+277,81:POKECO+277,0:K=277:J=32
1115 RETURN
1120 POKES,222
1121 GOSUB2000
1122 POKECA+279,81:POKECO+279,0:K=279:J=32
1125 RETURN
1130 POKES,224
1131 GOSUB2000
1132 POKECA+170,81:POKECO+170,0:K=170:J=160
1135 RETURN
1140 POKES,225
1141 GOSUB2000
1142 POKECA+281,81:POKECO+281,0:K=281:J=32
1145 RETURN
1150 POKES,227
1151 GOSUB2000
1152 POKECA+172,81:POKECO+172,0:K=172:J=160
1155 RETURN
1160 POKES,229
1161 GOSUB2000
1162 POKECA+283,81:POKECO+283,0:K=283:J=32
1165 RETURN
1170 POKES,0:RETURN
1180 POKES,0:S=36876:RETURN
1190 POKES,0:S=36875:RETURN
1200 POKES,0:S=36874:RETURN
2000 POKECA+K,J:POKECO+K,0:RETURN
3000 DATA66,72,80,86,88,94,102,108,110,116,124,130,132,138
3001 DATA146,152,154,160,168,174,176,182,190,-1
3002 DATA68,70,74,76,78,82,84,90,92,96,98,100,104,106,112,114
3003 DATA118,120,122,126,128,134,136,140,142,144,148,150,156
3004 DATA158,162,164,166,170,172,178,180,184,186,188,192,194,-1

```

READY.

Controllo del cursore e videogiochi

Un sottoprogramma utile per generare videogames.

SCRIVIAMO malvolentieri articoli sui videogiochi perchè rappresentano il prodotto deteriorato di una malsana società consumistica, ma soprattutto per via che i nostri redattori non sono mai riusciti a conseguire un punteggio dignitoso (ma come fanno certi dannati ragazzini a raggiungere "score" incredibili?).

Scherzi a parte profitteremo anche stavolta per fare un po' di didattica, consentendo ai principianti di capire in che modo sia possibile, in un videogioco, spostare lungo lo schermo un... qualcosa (pedina o cursore).

Un cenno alle locazioni di schermo

Ad ogni carattere visualizzabile sullo schermo, corrisponde una locazione di memoria, nella quale è possibile "scrivere" il carattere stesso. Ne deriva che le locazioni di schermo sono 506, valore questo, dato dal prodotto di 22 colonne per 23 righe. In effetti la quantità di memoria RAM necessaria per la gestione dello schermo è esattamente il doppio di 506. Associate alle locazioni di schermo sono, infatti, da considerare altre 506 locazioni, dette,

Tabella 1

Indirizzi delle locazioni richiamate nell'articolo.

Config. Locazioni	Inespanso	+3K	+8K in poi
Basic	4096-7679	1024-7679	4608 in poi
Schermo	7680-8191	7680-8191	4096-4607
Colore	38400-38911	38400-38911	37888-38399

per brevità, "di colore".

Il carattere di schermo come vedremo tra breve, deve essere "colorato" affinché si renda visibile. Inoltre i cosiddetti indirizzi delle locazioni di schermo e colore (rappresentate da due blocchi di 506 locazioni contigue RAM) cambiano a seconda della eventuale presenza di espansioni di memoria RAM (vedi tabella 1).

Da un attento esame ci accorgiamo anzitutto che la differenza dei valori estremi di schermo (es. 8191-7680) risulta essere 511 e non 506. Il motivo di questa diversità esula dallo scopo del pre-

sente articolo e sarà riproposto in altra occasione. Le considerazioni che seguono, terranno conto pertanto dei primi 506 byte di schermo e, ovviamente, dei corrispondenti 506 byte di colore. Dalla tabella si nota, inoltre, che le locazioni di schermo sono in coda a quelle dedicate ai programmi Basic nei casi di Vic 20 in versione base (inespanso) ed in quella con la scheda da 3K, mentre sono ubicate prima dell'area Basic, nel caso di Vic dotati di cartuccia da 8K oppure 16K di RAM. Il motivo del cambio di indirizzo, che è realizzato in modo completamente automatico al-

l'atto dell'accensione del computer, è da ricercarsi, come vedremo in altri articoli, nell'ottimizzazione della memoria disponibile.

Prima fase per la comprensione del listato

Eseguite nell'ordine i passi seguenti:

1/ accendete il computer dotato, oppure no, di un'espansione di memoria.

2/ Cancellate lo schermo (tasti SHIFT + CLR/HOME).

3/ Premete due volte il tasto Return.

4a/ Digitate:

A=7680: B=38400

se non avete espansioni o se avete inserito quella da 3K.

4b/ digitate:

A=4096 : B=37888

nel caso in cui abbiate inserito l'espansione RAM da 8 oppure 16K.

5/ Digitate:

POKE A,0: POKE B,0

Se avete seguito (ALLA LETTERA) le semplici indicazioni dovrete a questo punto vedere nell'angolo in alto a sinistra il carattere @ (gli inglesi lo leggono, at mentre in Italia è invalso l'uso di chiamarlo "chiocciola"), scritto in nero.

6/ Digitate:

POKE A+1,0

Questa operazione dovrebbe, a rigor di logica, fare apparire lo stesso carattere @ nella locazione successiva a quella numerica con A (seconda cella in alto a sinistra). Notiamo con sorpresa che ciò non accade. Ma se noi... 7/... saliamo in alto col cursore (tasto HOME) e lo posizioniamo nella seconda cella in alto a sinistra noteremo che il carattere @ viene evidenziato ad ogni lampeggio del cursore.

8/ Torniamo, grazie al tasto CRSR (quella per gli spostamenti verticali) sul rigo del comando digitato al punto 6 e modifichiamolo in:

POKE A+1,0: POKE B+1,0

Compare, finalmente, la seconda chiocciolina.

L'esperimento appena visto dimostra che:

- è possibile fare apparire in qualsiasi locazione dello schermo, a patto ovviamente di calcolarne l'indirizzo, uno qualsiasi dei 256 caratteri Commodore (maiuscole, minuscole, caratteri grafici, caratteri reversati);

- è indispensabile, dopo aver trascritto il carattere nella locazione di schermo desiderata, colorare la cella stessa, servendosi del comando POKE che agisce sulla cosiddetta mappa del colore schermo. Si fa notare che se conferiamo al carattere visualizzato lo stesso colore del fondo, il carattere stesso risulta invisibile benchè... presente. Esempio: rosso su rosso, verde su verde, ecc.

- quando si digita un comando

del tipo:

- PRINT "PROVA"

il computer non solo provvede a far apparire i caratteri desiderati, ma colora automaticamente le celle di memoria destinate ad ospitare i caratteri stessi. Ricorrendo, invece, ad operazioni di POKE che interessano l'area di schermo è come se perdessimo tale automatismo, e quindi siamo costretti, per evidenziare un carattere, a colorarlo mediante un secondo comando di POKE agente sull'area colore;

- gli argomenti delle POKE di schermo non corrispondono sempre agli argomenti delle istruzioni PRINT CHR\$(X). Esempio: la battitura di

POKE A,80: POKE B,0

(intendendo con A e B le locazioni di schermo e del colore) porta alla visualizzazione di un carattere diverso da:

PRINT CHR\$(80)

- ricorrendo a PRINT non ci si deve preoccupare di indicare le locazioni di schermo che saranno interessate, mentre, utilizzando i POKE di schermo e di colore, è indispensabile controllare attentamente gli indirizzi prima di eseguire il comando.

Una locazione interessante: la 203

Tutte le volte che premiamo un tasto del Vic 20, il Sistema Operativo (O.S.) del computer

provvede a individuarlo, a codificarlo con un valore intero compreso tra 0 e 255 e a memorizzarlo nella locazione di memoria 203. Per renderci conto di quanto asserito, facciamo girare a parte il microprogramma riportato nella riga 150 del listato pubblicato:

```
150 PRINT PEEK(203):  
GOTO 150
```

Non appena digitiamo RUN noteremo che sullo schermo compare in continuazione il numero 64. Questo è infatti, il codice del Vic che indica all'O.S. che nessun tasto viene premuto in quel momento. Proviamo ora a premere un tasto qualunque e teniamolo pigiato: il numero 64 sarà immediatamente sostituito da un nuovo valore che rappresenta il codice relativo al tasto premuto. La riga 150 serve dunque per esaminare i codici di ciascun tasto del Vic.

Che cosa fa il programma

Il programma, per lo scopo didattico per cui è stato scritto, si limita a far spostare, sullo schermo del Vic, ad una velocità stabilita, un carattere controllato dalla pressione di alcuni tasti:

Y	movimento verso l'alto
B	“ “ il basso
G	“ “ sinistra
J	“ “ destra

Il carattere che viene spostato sullo schermo non è sempre lo stesso. Infatti nei movimenti verso il basso assume la forma di una "V", in quelli verso l'alto di "∧", verso destra diventa ">" e verso sinistra prende la forma di "<". Ho pensato, in tal modo, di conferire maggior... realismo al movimento e di dare un piccolo suggerimento per sofisticazioni da introdurre.

Come funziona

Volendo, il carattere che si muove, può lasciare dietro a sé una scia. Se infatti alla domanda CARATTERE? (Riga 180) rispondiamo con 32, il carattere si muoverà lungo lo schermo senza lasciare apparentemente alcuna traccia (in realtà viene invece trascritto il carattere di spazio bianco). Al valore 160 corrisponderà una scia nera e ad altri valori, compresi ovviamente tra 0 e 255, corrisponderanno i vari caratteri tipici del Vic, da individuare grazie alla riga 150.

Riga 200-210. A seconda della espansione posseduta e controllando la tabella 1, il lettore trascriverà la riga 200 (cancellando REM) oppure la 210.

Riga 220. Calcolo dei blocchi di memoria RAM dedicati allo schermo e al colore.

Riga 230. Invece di provvedere, nel seguito del programma, a colorare una cella video, dopo aver depositato un carattere, si è preferito colorare ciascuna cella dello schermo fin dall'inizio, in mo-

do da snellire il programma stesso.

Righe 240-290. Esame dell'eventuale tasto premuto. In caso di pressione di tasti diversi da B, Y, G, J o di nessun tasto, si ritorna continuamente (riga 290) alla riga 240.

Righe 310-320. Spostamento in basso del carattere. Prima di spostarlo è però necessario controllare che non si esca dalla mappa del video (riga 310). Se infatti il carattere, al momento della pressione del tasto B, è già presente sull'ultima riga, non ha senso dare il comando di un'ulteriore discesa.

Righe 340-350. Controllo salita carattere.

Righe 370-380. Spostamento a destra. La cella interessata è individuata dal valore S2 (numero di cella contenente il carattere PRIMA della pressione di J) incrementato di un'unità; il carattere presente nella cella video, prima della pressione di J, deve essere sostituito dal carattere scelto alla riga 180. Infine (riga 380) il valore di S2 viene "aggiornato" per le successive elaborazioni.

Righe 400-410. Spostamento a sinistra.

Riga 440. La variabile LO, definita alla riga 170, consente di introdurre un ritardo nello spostamento del carattere tra una cella e la successiva. Se infatti alla domanda LIVELLO? rispondiamo col valore nullo, la velocità di spostamento è piuttosto elevata ed è necessaria una certa prontezza di riflessi per il controllo del cursore.

Quello proposto, come già detto, deve essere considerato come sottoprogramma, magari da sofisticare, in listati di videogio-

chi, in cui è necessario spostare una pedina sullo schermo. Un piccolo suggerimento prima di concludere: dopo aver trascritto

il programma dalla rivista, inserite la riga 190, sostituite la 240 con l'altra riportata in fondo al listato e... buon divertimento! ■

```
100-REM *** CONOSCERE MEGLIO IL VIC 20 ***
110 REM ***
120 REM *** ROUTINE FONDAMENTALI PER ***
130 REM *** LA GENERAZIONE DI GIOCHI ***
140 :
150 REM PRINT PEEK (200): GOTO 150: ESAME CODICE CARATTERE
160 :
170 INPUT"LIVELLO (0-1000)": LOOP: REM DIFFICOLTA' = RITARDO
180 INPUT"CARATTERE (0-255)":CA: IF CA>255 THEN 180: REM CARATTERE... "SCIA"
190 :
200 REM S1 = 7680: C1 = 38400: REM VIC INESPANSO
210 S1 = 4096: C1 = 37338: REM VIC ESPANSO (DA BK IN SU)
220 S2=S1: S3=S1+22*23-1: C2=C1+22*23-1: PRINT "Q"
230 FOR X=C1 TO C2 :POKE X,0: NEXT: REM COLORAZIONE FONDO SCHERMO
240 X = PEEK (200): REM ESAME TASTO EVENTUALMENTE PREMUTO
250 IF X=35 THEN GOSUB 310: GOSUB 440: GOTO 240: REM TASTO B
260 IF X=11 THEN GOSUB 340: GOSUB 440: GOTO 240: REM TASTO Y
270 IF X=19 THEN GOSUB 400: GOSUB 440: GOTO 240: REM TASTO G
280 IF X=20 THEN GOSUB 370: GOSUB 440: GOTO 240: REM TASTO J
290 GOTO 240: REM R. DE SIMONE SOFTWARE '83
295 :
300 REM MOVIMENTO VERSO IL BASSO. 22 = CARATTERE v
310 IF S2+22 > S3 THEN RETURN
320 POKE S2+22,22: POKE S2,CA :S2 = S2+22: RETURN
325 :
330 REM MOVIMENTO IN ALTO. 30 = CARATTERE ↑
340 IF S2-22 < S1 THEN RETURN
350 POKE S2-22,30: POKE S2,CA :S2 = S2-22: RETURN
355 :
360 REM MOVIMENTO A DESTRA. 62 = CARATTERE >
370 IF S2+1 > S3 THEN RETURN
380 POKE S2+1,62: POKE S2,CA :S2 = S2+1: RETURN
385 :
390 REM MOVIMENTO A SINISTRA. 60 = CARATTERE <
400 IF S2-1 < S1 THEN RETURN
410 POKE S2-1,60: POKE S2,CA: S2 = S2-1: RETURN
420 :
430 REM SOTTOPROGRAMMA DI RITARDO
440 FOR I=1 TO LOOP: NEXT: RETURN

READY.

190 A(1)=35: A(2)=11: A(3)=19: A(4)=20

240 X = A(INT(RND(J) * 4))
```

DAL N°1:

Perchè accontentarsi di un videogame?



Speciale, specialissimo!
Invece dei soliti videogiochi prova VIC 20,
e guarda quante cose fa in più!

1. VIC 20 ha una valanga di videogiochi, uno più bello dell'altro, uno più nuovo dell'altro.
2. Ma VIC 20 è un computer e fa molto di più.
3. Lo usi per la scuola, o per la casa, o per la professione. Ci sono, pronti pronti,

un mucchio di programmi. Metti le cassette
e via con cose utili.

4. Puoi imparare il BASIC, la lingua del futuro (ed è facile facile imparare a programmare).

5. Nel mondo sono stati venduti più di un milione di VIC, a gente sveglia, quelli del 2000.

VIC 20

**Il più venduto nel mondo.
a 199.000 Lire!**
più IVA

ETHOS



6. VIC 20 ora costa solo 199.000 lire
più IVA. Da sballo, no?
Perchè accontentarsi di un semplice
videogioco?

Commodore Italiana S.p.A.
Via F.lli Gracchi 48 - Cinisello Balsamo (MI)
Tel. 02/6125651-6123253

commodore
COMPUTER

Ed ora passo al Pet

Come leggere col Pet (4000 8000) i programmi scritti con i computer Vic 20 e Commodore 64.

DOVREBBE essere noto che i computer dotati di Basic Microsoft, e in particolare i Commodore, utilizzano aree di memoria ben definite per deporre i programmi dell'utente. L'indirizzo di partenza dei listati è contenuto in due locazioni di memoria che rappresentano, rispettivamente, la parte bassa e quella alta dell'indirizzo stesso.

Negli elaboratori PET (qualsiasi serie) e nel VIC 20, dotato, però, dell'espansione di memoria di 3K di RAM, i programmi Basic vengono posizionati a partire dalla locazione 1024; nel Vic senza espansioni, dalla locazione 4096, nel Vic con espansioni da 8 e più K, dalla locazione 4608 e nel Commodore 64, da 2048.

Quando impartiamo l'ordine SAVE, l'inizio del file-programma contiene l'indirizzo di partenza in cui è allocato il programma che sta per essere registrato. Quando invece diamo il comando LOAD, il comportamento dei PET si diversifica da quello dei VIC (20 e 64).

Nel primo caso, infatti, il PET carica, *sempre e comunque*, il programma considerando l'indirizzo posto in testa al file.

Nei casi del Vic 20 e del Commodore 64, invece, il programma viene posizionato automaticamente a partire dalla locazione in cui il computer stesso, nella particolare configurazione di memoria adoperata, allocherebbe un programma se questo fosse digi-

tato normalmente da tastiera.

In altre parole, un programma registrato in un momento in cui il Vic 20 era senza espansioni di memoria, (locazione inizio Basic = 4096), verrà posizionato, se riletto con espansione di memoria da 8K, a partire da 4608.

L'utente non si accorge affatto del cambiamento, proprio grazie all'automatismo della procedura.

Sembrirebbe, pertanto, che non è possibile caricare su PET un programma scritto dai suoi... fratelli minori. Naturalmente, siamo qui per illustrarvi la possibilità di riuscire nell'intento, a patto di osservare *alla lettera* la procedura qui di seguito riportata [(R) = battere il tasto di Return]:

- 1/ caricare sul PET il programma in oggetto col "solito" LOAD;
- 2/ quando l'operazione è completata, *non listate* il programma per nessun motivo;
- 3/ entrate in monitor digitando: SYS (1024) (R);
- 4/ visualizzate il rigo di otto byte battendo: M 0400 0401 (R);
- 5/ ignorate il contenuto delle otto locazioni di memoria visualizzate dopo la fase 4/;
- 6/ servendosi dei tasti di controllo cursore, modificate le locazioni specificate come segue:

• caso di programma registrato con Vic 20 senza alcuna espansione di memoria:

0400 : 00 01 10 00 00 00 00
00 (R);

• Caso di Vic 20 con espansione da 8 o più K RAM:

0400 : 00 01 12 00 00 00 00
00 (R);

• caso di Commodore 64:

0400 : 00 01 08 00 00 00 00
00 (R);

7/ "uscire" dal monitor digitando "X" (R);

8/ *non* chiedere il listato in questo momento: il PET andrebbe in Tilt;

9/ digitare:

0 (R).

Comportandovi, cioè, come se voleste cancellare, da un normalissimo programma Basic, la linea numerata con zero.

A questo punto il programma è posizionato correttamente in memoria: potete listarlo, usarlo, modificarlo, ecc. Nel caso abbiate un programma di cui non si conosce il computer di provenienza, l'unico modo è quello di procedere per tentativi fino a che non appare il listato. Si tenga presente che, come già detto, se il programma proviene da un Vic 20, dotato di 3K RAM, non è necessaria alcuna modifica. Se, invece, nonostante il procedimento suggerito, non appare il listato o il PET si "impianta" ad ogni tentativo, vorrà dire che il programma in oggetto è scritto non in Basic, ma in Linguaggio Macchina, oppure, è protetto. In questo caso non c'è nulla da fare... ■

Riservato
agli ingegneri

Il miglior software tecnico su elaboratori CBM - Commodore ora anche disponibile su Commodore 64

“S.S. - 80”

L'ormai famoso programma per il calcolo delle strutture intelaiate piane in c.a., in zona sismica, che sviluppa e disegna anche le carpenterie delle armature.
(Ultima versione Luglio/1982 nostra esclusiva).

“FONDAZIONI”

Risolve tutti i problemi di fondazioni (trave elastica su suolo elastico) di strutture in c.a. in zona sismica e non, risolvendo l'intero graticcio di fondazione e proponendo una carpenteria sofisticata ed ottimizzata.

“MURI DI SOSTEGNO”

A gravità, a mensola o a contrafforti, anche in zona sismica, secondo il D.M. del 21/1/1981.

“PENDII”

Analizza la stabilità di un pendio o di un fronte di scavo sotto diverse condizioni e la verifica relativa viene condotta in termini di tensioni effettive; la stima dei fattori di sicurezza viene effettuata secondo i metodi di Fellenius, Bishop e Jambu.

“COMPUTI METRICI”

Analisi ed elenco prezzi Metodo veloce e completamente automatizzato per il computo e la stima dei lavori.

“REVISIONE PREZZI”

Secondo le disposizioni di legge vigenti. Praticità ed automazione consentono di eseguire velocemente revisioni di prezzi anche per lunghi periodi.

Richiedeteci documentazione e output dei programmi di vostro interesse. Resterete sbalorditi dalla versatilità e dalla completezza del nostro software.

SIRANGELO COMPUTER Srl Via Parisio 25 - Cosenza 0984-75741

NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW

È pronto il nuovissimo programma

“ORARIO SCOLASTICO”

DAL N°1: COM



Mai un grande perso

Quest'anno, fatti un regalo intelligente: un computer dalle caratteristiche incredibili. Vediamole.

1. Commodore 64 è potente, sofisticato, professionale.

2. Ha una vastissima gamma di programmi già pronti, lo usi nella professione, a casa, a scuola, nella ricerca scientifica, con facilità e totale affidabilità.

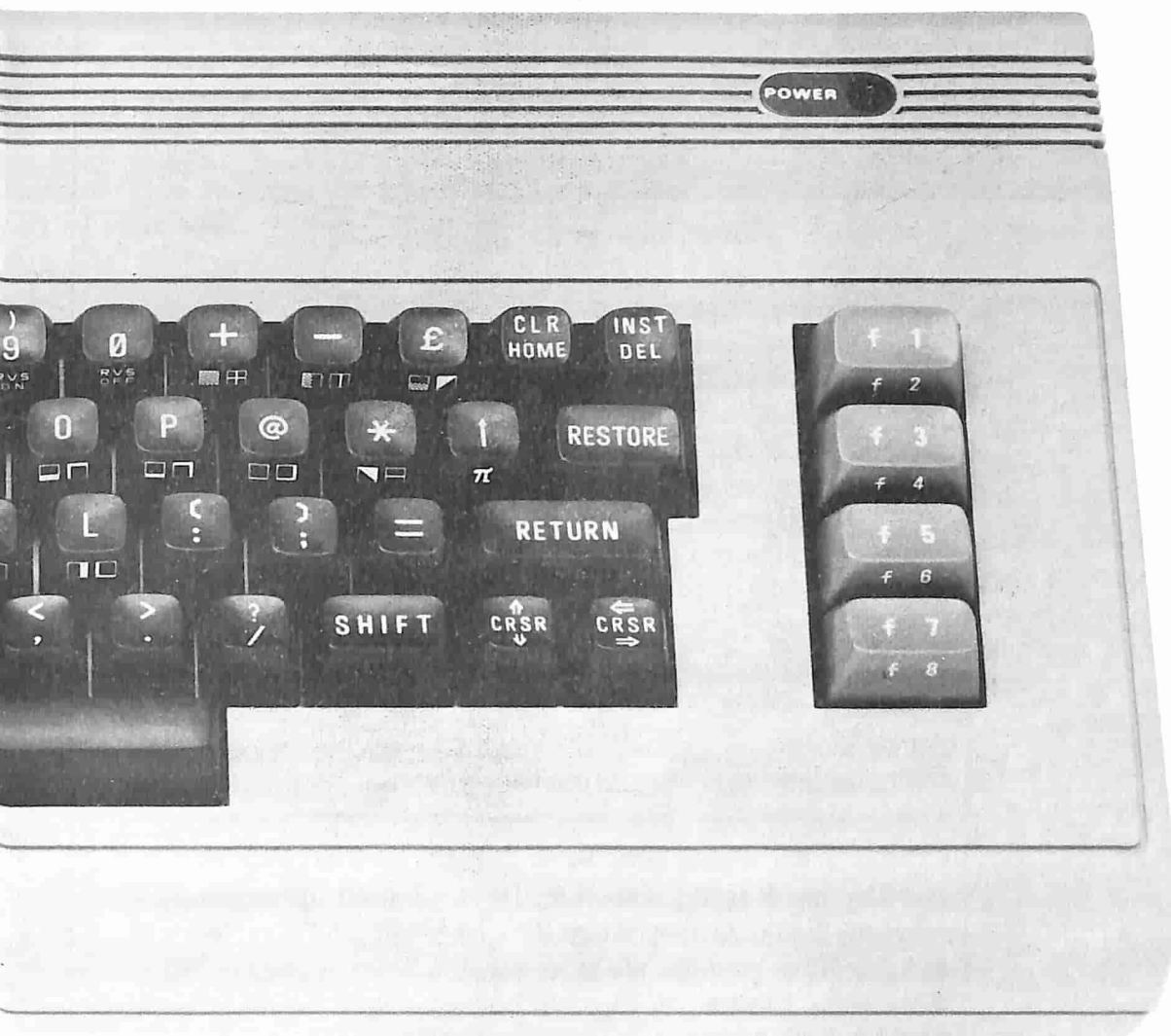
3. Ha un'incredibile memoria (64 K), un sintetizzatore sonoro professionale, produce effetti tridimensionali.

4. Ti diverti perchè è anche un sofisticato videogioco.

5. Con Commodore 64 entri nel futuro, tasto dopo tasto.

6. Commodore 64 oggi lo puoi avere a prezzo davvero speciale:

MODORE 64



nal è costato così poco

approfittane però perchè sta andando a ruba,
e chi primo arriva...

Vieni a un punto vendita Commodore:
ti aspetta una bella sorpresa.

Commodore Italiana S.p.A.
Via F.lli Gracchi 48 - Cinisello Balsamo (MI)
Tel. 02/6125651-6123253

 **commodore**
COMPUTER

Stampa in corsivo

Una scrittura "personalizzata" con qualsiasi sistema Commodore.

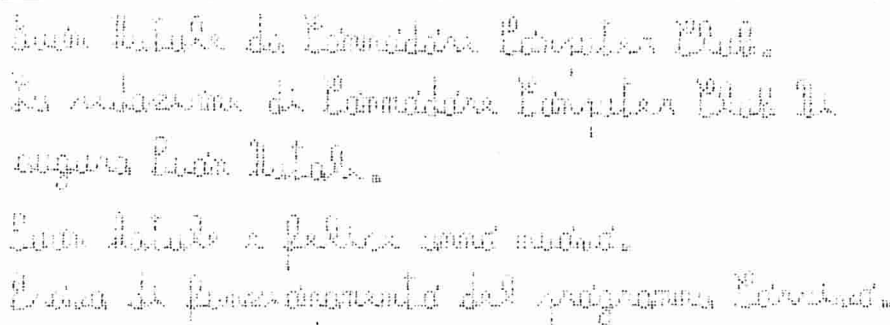
Di Bellù Giovanni

IL programma, che gira su configurazione Pet + stampante 4022 (o 3022) e di cui viene data anche la versione per il Vic con almeno l'espansione di 8K, come già indica il nome, serve a fare scrivere il computer in corsivo, cioè con quel tipo di scrittura che noi abitualmente usiamo quando scriviamo a mano.

Come si usa

All'inizio bisogna attendere un po' - circa mezzo minuto - per permettere al computer di leggere ed elaborare tutti quei numeri (circa 1500!) che sono scritti nei DATA. Quindi, appare sul video la scritta "INTRODUCI IL TESTO", ed una linea lunga 79 caratteri - questa è la lunghezza massima di una riga del testo da fare stampare.

L'input del testo avviene mediante l'uso dell'istruzione GET, il che si è reso necessario per controllare i dati introdotti; e che per il resto è uguale all'input: si può, cioè, usare il tasto DELETE, e bisogna premere RETURN alla fine della riga.



Buon Natale da Commodore Computer Club.
La redazione di Commodore Computer Club ti
augura Buon Natale.
Buon Natale e felice anno nuovo.
C'è un grande finanziamento del programma "Corsivo".

Esempio di stampa ottenibile col programma proposto.

In seguito si può introdurre un'altra riga di testo, fino ad un massimo di 50 (Questo valore si può cambiare modificando il valore della DIMB\$(50) di riga 270).

Per indicare che si è finito di introdurre il testo, bisogna premere "*" all'inizio di una nuova riga. Se questo carattere viene usato non in prima posizione, esso viene accettato e naturalmente stampato. A questo punto comincia la stampa.

Per quanto riguarda i caratteri che si possono usare, essi sono:

- tutte le lettere dell'alfabeto latino sia minuscole che maiuscole (minuscole shiftate);

- tutti i caratteri numerici;
- i caratteri di punteggiatura: .,:;!;
- altri caratteri che in ascii vanno da 32 (spazio) al 64 (a commerciale).

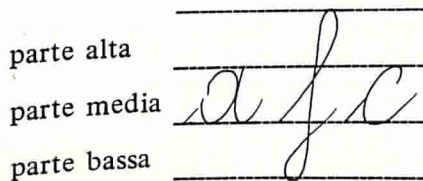
Tutti gli altri caratteri non vengono accettati. Una volta copiato il listato potrete scrivere tutto quello che volete e, dato che siamo in clima natalizio, potete mandare le vostre lettere di auguri scritte dal Computer...

Come funziona

Tutti quei numeri troppi?) nei DATA servono per la formazione dei caratteri speciali che si possono formare usando una del-

le caratteristiche della stampante Commodore.

Il metodo usato per la formazione dei caratteri consiste nello scomporre ogni lettera in tre parti: alta, media, bassa come in figura:



Per questa ragione per ogni lettera ci sono tre caratteri speciali per cui ci sono 18 dati per ciascuna lettera (6 per carattere per tre caratteri).

Nella fase di stampa, per ogni riga di testo, verranno stampate tutte le parti "alte" delle lettere, poi quelle "medie", ed infine quelle "basse", per cui ogni riga di testo occupa tre righe di stampa.

La versione VIC

Il procedimento di formazione dei caratteri è il medesimo, con l'unica differenza che, mentre nel Pet la formazione dei 6 numeri per ogni carattere speciale avviene tenendo conto delle potenze di 2 a partire dal 6 fino allo 0, nel VIC è esattamente l'opposto, come in figura.

Pet: 2 elev. 6
2 elev. 5

Vic: 2 elev. 0
2 elev. 1

...

...

2 elev. 0

2 elev. 6

Per cui si è reso necessario, per non rifare la mappa caratteri, fare una "inversione". Questo avviene nelle linee di programma da 10000 a 10012, in cui il numero viene trasformato in binario, quindi, "ribaltato" e ritrasformato in decimale.

Altre modifiche riguardano la parte di stampa:

+ 250 per i diversi OPEN e istruzioni grafiche;

+ 300 per la formazione dei caratteri su VIC bisogna sommare 128 al numero nei DATA;

+ 1280 e 1290 per le istruzioni PRINT#.

Per il resto la versione VIC è esattamente identica a quella per il Pet.

Programma "crea"

Questo programmino si è rivelato molto utile per la definizione dei caratteri.

All'inizio chiede quale carattere dobbiamo formare, poi sullo schermo appaiono tre rettangoli di 6*7, uno sopra l'altro, che rappresentano le tre parti in cui viene divisa la lettera (vedi prima). In fondo allo schermo appare il cursore lampeggiante (in input).

A questo punto bisogna sola-

mente salire con gli appositi tasti di controllo del cursore e mettere il carattere "Q shiftato" nelle posizioni in cui sulla stampante deve venire stampato un punto.

Finito di formare il carattere, come meglio preferiamo, basta premere RETURN su una linea in cui vi siano dei caratteri (non importa quali, e questo lo potrete capire facilmente analizzando il programma o, meglio ancora, facendolo girare.

Quindi il computer analizza i tre rettangoli di prima e automaticamente visualizza sullo schermo il numero di linea corrispondente a quel carattere seguito da un DATA che contiene i 18 dati trovati, ed infine, un'istruzione REM seguita dal carattere in questione, e sotto a questa linea c'è la scritta RUN.

A questo punto dobbiamo premere RETURN 2 volte; una per memorizzare la linea che contiene i dati del carattere in questione, l'altra per ritornare nel programma.

Nel frattempo, il carattere che abbiamo formato viene stampato, per cui possiamo verificare subito se c'è qualche cosa da modificare o se va bene. ■

Bellù Giovanni

C O R S I V O
Di Bellù Giovanni - Nov. 1983 -

```
100 REM *****
110 REM **          C O R S I V O          **
120 REM *****
```



```

130 REM ** VERSIONE PET+4022 (3022) **
140 REM ****
150 REM **      BY BELLU' GIOVANNI      **
160 REM ****
170 REM **      VIA GIARDINI N. 20      **
180 REM ****
190 REM **      20038 SEREGNO          **
200 REM ****
210 REM **      TELEFONO: 0362/239580   **
220 REM ****
230 REM **      - NOVEMBRE 1983 -      **

```

```

240 REM ****
250 OPEN6,4,6:PRINT#6,CHR$(22):OPEN3,4:OPEN5,4,5:POKE59468,14
260 PRINT"ATTENDEI STO MEMORIZZANDO I DATI "
270 DIMA$(85,2),A(18),B$(50)
280 FORK=1TO85
290 FORJ=1TO18:READA(J):NEXTJ
300 FORH=0TO2:FORJ=1TO6:A$(K,H)=A$(K,H)+CHR$(A(J+6*H)):NEXTJ,H,K
310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:REMSPACE
320 DATA0,0,127,127,0,0,0,0,123,123,0,0,0,0,0,0:REM
330 DATA8,16,96,8,16,96,0,0,0,0,0,0,0,0,0,0:REM"
340 DATA0,0,0,0,0,0,20,127,20,127,20,0,0,0,0,0:REM#

350 DATA96,24,4,63,68,56,6,41,41,127,36,34,0,0,0,0,0:REM$
360 DATA1,1,1,0,0,1,99,36,104,23,37,71,0,0,0,0,0:REM%
370 DATA1,2,2,2,1,0,102,25,9,21,114,5,0,0,0,0,0:REM&
380 DATA0,0,6,28,112,0,0,0,0,0,0,0,0,0,0,0:REM'
390 DATA0,31,32,64,64,64,0,124,2,1,1,1,0,0,0,0,0:REM(
400 DATA64,64,64,32,31,0,1,1,1,2,124,0,0,0,0,0:REM)
410 DATA0,1,3,3,1,0,0,64,96,96,64,0,0,0,0,0:REM*
420 DATA1,1,7,7,1,1,64,64,112,112,64,64,0,0,0,0,0:REM+
430 DATA0,0,0,0,0,0,0,3,3,0,0,0,4,24,96,0,0:REM,
440 DATA1,1,1,1,1,1,64,64,64,64,64,64,0,0,0,0,0:REM-
450 DATA0,0,0,0,0,0,0,7,7,7,0,0,0,0,0,0:REM.
460 DATA1,1,25,25,1,1,64,64,76,76,64,64,0,0,0,0,0:REM/
470 DATA31,32,64,64,32,31,124,2,1,1,2,124,0,0,0,0,0:REM0
480 DATA8,16,32,64,127,0,0,1,1,1,127,1,0,0,0,0,0:REM1
490 DATA48,64,64,65,66,60,28,34,65,1,1,1,0,0,0,0,0:REM2
500 DATA32,65,65,65,62,0,2,65,65,65,62,0,0,0,0,0:REM3
510 DATA0,0,1,6,24,96,0,60,68,4,31,4,0,0,0,0,0:REM4
520 DATA0,127,65,65,64,96,2,1,1,1,126,0,0,0,0,0:REM5
530 DATA7,8,16,32,64,64,124,2,33,33,33,30,0,0,0,0,0:REM6
540 DATA112,64,64,65,78,112,0,67,92,96,64,64,0,0,0,0,0:REM7
550 DATA28,34,65,65,34,28,60,66,1,1,66,60,0,0,0,0,0:REM8
560 DATA60,66,66,66,63,0,6,1,1,2,124,0,0,0,0,0:REM9
570 DATA0,0,0,0,0,0,0,99,99,0,0,0,0,0,0,0:REM:
580 DATA0,0,0,0,0,0,0,51,51,0,0,0,4,24,96,0,0:REM;
590 DATA0,0,0,0,1,1,24,24,36,102,67,1,0,0,0,0,0:REM<
600 DATA6,6,6,6,6,6,48,48,48,48,48,48,0,0,0,0,0:REM=

```



```

610 DATA1,1,0,0,0,0,1,67,102,36,24,24,0,0,0,0,0:REM>
620 DATA24,32,64,65,66,60,0,0,125,0,0,0,0,0,0,0:REM?
630 DATA0,0,0,0,0,0,62,65,93,85,84,60,0,0,0,0,0:REMO
640 DATA0,0,0,0,0,0,2,62,65,65,62,7,0,0,0,0,0:REMA
650 DATA0,31,32,32,31,0,2,124,31,33,78,10,0,0,0,0,0:REMB
660 DATA0,0,0,0,0,0,2,28,34,65,65,2,0,0,0,0,0:REMC
670 DATA0,0,0,0,127,0,2,30,33,33,127,6,0,0,0,0,0:REMD
680 DATA0,0,0,0,0,2,52,72,52,2,2,0,0,0,0,0:REME
690 DATA0,31,32,32,31,0,2,127,13,50,66,2,0,127,0,0,0:REMF
700 DATA0,0,0,0,0,0,2,62,65,65,63,3,0,28,34,66,126,0:REMG
710 DATA0,31,32,32,31,0,2,127,10,52,98,3,0,0,0,0,0:REMH
720 DATA0,0,2,0,0,0,2,4,120,12,4,2,0,0,0,0,0:REMI
730 DATA0,0,0,0,1,0,3,0,0,0,63,1,0,112,0,8,112,0:REMJ
740 DATA0,0,0,0,0,0,2,127,8,20,34,66,0,0,0,0,0:REMK
750 DATA0,31,32,32,31,0,2,4,120,24,100,2,0,0,0,0,0:REML
760 DATA0,0,0,0,0,0,126,64,126,64,124,2,0,0,0,0,0:REM M
770 DATA0,0,0,0,0,0,62,64,62,64,60,2,0,0,0,0,0:REM N
780 DATA0,0,0,0,1,6,2,62,65,65,126,2,0,0,0,0,0:REMO
790 DATA0,0,0,0,0,0,2,12,16,96,127,2,0,0,0,0,127,0:REMP
800 DATA0,0,0,0,0,0,2,30,33,65,127,2,0,0,0,0,127,0:REM Q
810 DATA0,0,0,0,0,0,2,12,112,64,120,6,0,0,0,0,0:REMR
820 DATA0,0,0,0,0,0,2,12,48,66,66,62,0,0,0,0,0:REMS
830 DATA0,16,127,16,16,0,2,4,127,1,2,2,0,0,0,0,0:REMT
840 DATA0,0,0,0,0,0,2,126,1,1,126,2,0,0,0,0,0:REMU
850 DATA0,0,0,0,0,0,30,16,15,1,14,18,0,0,0,0,0:REMV
860 DATA0,0,0,0,0,0,2,127,2,4,2,127,0,0,0,0,0:REMW
870 DATA0,0,0,0,0,0,99,20,8,20,99,2,0,0,0,0,0:REMX
880 DATA0,0,0,0,0,0,2,126,1,2,127,2,0,3,12,48,64,0:REMY
890 DATA0,0,0,0,0,0,14,112,70,74,82,98,0,0,0,0,0:REMZ
900 DATA0,31,32,32,31,0,2,126,1,1,126,1,0,0,0,0,0:REMA
910 DATA0,127,65,66,60,0,2,127,1,65,62,2,0,0,0,0,0:REMB
920 DATA96,27,4,58,66,60,2,126,1,1,3,2,0,0,0,0,0:REMC
930 DATA0,127,64,48,15,0,2,127,1,2,124,2,0,0,0,0,0:REMD
940 DATA64,48,15,52,36,24,2,6,73,41,17,18,0,0,0,0,0:REME
950 DATA120,64,67,76,112,64,66,71,97,81,79,2,0,0,0,0,0:REMF
960 DATA64,51,28,40,72,112,2,124,3,1,15,3,6,9,17,34,124,0:REMG
970 DATA0,127,1,1,127,0,2,126,0,0,126,2,0,0,0,0,0:REMH
980 DATA112,64,64,67,124,0,2,14,1,97,30,2,0,0,0,0,0:REMI
990 DATA0,0,0,0,127,0,2,1,1,1,126,0,0,0,0,0,0:REMJ
1000 DATA0,127,3,4,8,112,2,127,0,64,48,15,0,0,0,0,0:REMK
1010 DATA64,56,7,60,68,56,2,5,127,8,4,2,0,0,0,0,0:REML
1020 DATA127,64,127,64,127,0,126,0,126,0,120,6,0,0,0,0,0:REMM
1030 DATA48,64,63,64,127,0,2,4,126,0,124,2,0,0,0,0,0:REM N
1040 DATA0,63,64,64,63,64,2,126,1,1,126,0,0,0,0,0,0:REMO
1050 DATA0,127,65,64,63,0,2,127,65,65,65,2,0,0,0,0,0:REMP
1060 DATA0,63,64,64,64,63,2,126,1,9,5,122,0,0,0,0,0:REM Q
1070 DATA0,127,64,65,62,0,2,127,64,32,28,2,0,0,0,0,0:REMR
1080 DATA96,16,60,75,72,48,6,9,1,65,57,6,0,0,0,0,0:REMS
1090 DATA96,64,65,70,120,0,2,60,70,1,1,6,0,0,0,0,0:REMT
1100 DATA0,127,0,0,127,0,2,126,1,1,126,2,0,0,0,0,0:REMU
1110 DATA56,64,127,0,31,32,2,2,126,1,127,2,0,0,0,0,0:REMV
1120 DATA0,127,0,0,0,127,2,127,6,56,6,127,0,0,0,0,0:REMW

```



```

1130 DATA112,14,1,14,112,0,3,28,96,28,2,2,0,0,0,0,0,0:REM X
1140 DATA64,56,7,0,127,0,2,1,2,92,100,4,0,0,0,0,0,0:REM Y
1150 DATA96,64,64,67,76,112,67,77,113,65,65,7,0,0,0,0,0,0:REM Z
1160 PRINT"□"
1170 PRINT"██████ INTRODUCI IL TESTO "
1180 FORK=0TO999:NEXT
1190 GOSUB1360
1200 FORI=0TOW-1
1210 FORJ=0TO2
1220 FORK=1TOLEN(B$(I))
1230 C#=MID$(B$(I),K,1)
1240 A=ASC(C#)
1250 IFA>31ANDR<91THENA=A-31:GOTO1280
1260 IFA>192ANDR<219THENA=A-192+59:GOTO1280
1270 PRINT"████ ERRORE " :END
1280 PRINT#5,A$(R,J):PRINT#3,TAB(K)CHR$(254)CHR$(141):NEXT
1290 PRINT#5,A$(1,1):PRINT#3,TAB(K)CHR$(254):NEXT
1300 A#="" :NEXT
1310 PRINT"████ FINE "
1320 PRINT"██████ ALTRO TESTO ?"
1330 GETA$:IFA#<>"S"ANDR#<>"N"THEN1330
1340 IFA#="N"THENEND
1350 W=0:GOTO1160
1360 P3=78:P4=2
1370 P#=""
1380 PRINT
1390 FORQ=1TOP3:PRINT"_" :NEXT:PRINT"| "
1400 FORQ=1TOP3+1:PRINT"███" :NEXT
1410 REM
1420 GETA$:IFA#<>" "THEN1450
1430 FORJ=1TO20:PRINT"███" :NEXT:GETA$:IFA#<>" "THEN1450
1440 FORJ=1TO10:PRINT"██ ███" :NEXT:GOTO1420
1450 IFA#=CHR$(13)THENPRINT"███" :GOTO1610
1460 IFASC(A#)=34THENPRINTR#A#CHR$(20)CHR$(148)"███" :P#=P#+A#:GOTO1410
1470 IFF#=""ANDR#="*"THENRETURN
1480 IFA#<>CHR$(20)GOTO1540
1490 PRINT"███"
1500 IFF#=""GOTO1410
1510 IFLEN(P#)=1THENP#="" :GOTO1530
1520 P#=LEFT$(P#,LEN(P#)-1)
1530 A#="███" :GOTO1590
1540 IFA#=CHR$(160)THENA#=" "
1550 IFASC(A#)>90ANDASC(A#)<193THEN1410
1560 IFASC(A#)>218THEN1410
1570 IFLEN(P#)=P3GOTO1620
1580 P#=P#+A#
1590 PRINTR#
1600 GOTO1410
1610 IFLEN(P#)<P4GOTO1410
1620 B$(W)=P#
1630 W=W+1:GOTO1360
READY.

```

La Redazione di *Commodore Computer Club*

suggerisce un felice anno nuovo

a tutti i suoi lettori

La memorizzazione dei dati nel III°

richiede alcuni minuti.

Attendete con pazienza !!!

La memorizzazione dei dati nel VIC 20

richiede alcuni minuti:

Attendete con pazienza !!

Esempi di stampa con la stampante 1515

Modifiche e aggiunte per la versione VIC 20

```
250 OPEN#4:PRINT#3,CHR$(6):PRINT CHR$(14):POKE36879,8:PRINT"#"
```

READY.

```
290 FORJ=1TO18:READA(J):GOSUB10000:NEXTJ  
300 FORH=0TO2:FORJ=1TO6:A$(K,H)=A$(K,H)+CHR$(A(J+6*H)+128):NEXTJ,H,K
```

READY.

```
1290 PRINT#3,A$(A,J):NEXT  
1290 PRINT#3,A$(1,1):NEXT
```

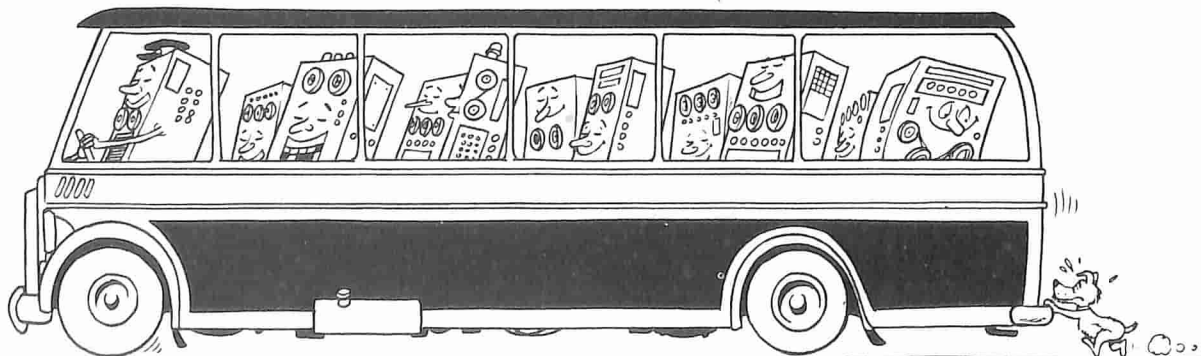
READY.

```
10000 OP=0:U$="":A$="":U=A(J):FORP=0TO6  
10002 IFU/2=INT(U/2)THENU$=U$+"0":GOTO10004  
10003 U$=U$+"1"  
10004 U=INT(U/2):NEXT  
10010 FORP=7TO1STEP-1:A$=A$+MID$(U$,P,1):NEXT  
10011 FORP=1TO7:IFMID$(A$,P,1)="1"THEN OP=OP+2+(P-1)  
10012 NEXT:A(J)=OP:RETURN
```

READY.

C R E A

```
100 REM *****
110 REM  *      ' C R E A '      *
120 REM  *****
130 REM  **PROGRAMMA PER CREARE**
140 REM  **I CARATTERI SPECIALI**
150 REM  **USATI NEL PROGRAMMA **
160 REM  **CORSIVO.          **
170 REM  **VERSIONE PET.     **
180 REM  *****
190 REM  ** BY BELLU' GIOVANNI **
200 REM  *****
210 REM  ** TEL. 0362/239580  **
220 REM  *****
230 PRINT "□":DIM A(18)
240 OPEN#5,4,5:OPEN#4,4:OPEN#6,4,6:PRINT#6,CHR$(25)
250 INPUT "□LETTERA:";L#
260 IF LEN(L#) <> 1 THEN 250
270 A=ASC(L#)
280 NL=1000+A
290 PRINT "□";
300 FOR K=1 TO 7:PRINT "TTTTTTT " :NEXT
310 FOR K=1 TO 7:PRINT "XXXXXXXXX " :NEXT
320 FOR K=1 TO 7:PRINT "LLLLLLL " :NEXT
330 INPUT A#
340 FOR K=0 TO 2:Q=K*7:FOR X=0 TO 5:Z=Z+1:FOR Y=0 TO 6
350 IF PEEK(32768+X+40*Y+40*Q)=81 THEN A(Z)=A(Z)+2*(6-Y)
360 NEXT Y:NEXT X:NEXT Q
370 PRINT "□";
380 PRINTNL "DATA";:FOR K=1 TO 18:PRINT RIGHT$(STR$(A(K)),LEN(STR$(A(K)))-1
390 PRINT CHR$(44);:NEXT:PRINT "III";CHR$(58)"REM" L#:PRINT "RUN"
400 PRINT "□"
410 FOR J=0 TO 2:A#="" :FOR K=1 TO 6:A#=A#+CHR$(A(K+6*J)):NEXT
420 PRINT#5,A#:PRINT#4,CHR$(254):NEXT
430 CLOSE#5:CLOSE#4
READY.
```



Guida mercato Commodore

Prodotto	Prezzo (IVA esclusa)
----------	-------------------------

VIC - 20

Home Computer Vic 20	199.000
Unità di espansione (1020)	295.000
Modulo di espansione (1023)	135.000
Cartuccia da 3K di memoria (1210)	66.000
Cartuccia da 8K di memoria (1110)	98.000
Cartuccia da 16K di memoria (1111)	172.000
Cartuccia Vic rel (4011)	95.000
Permette di controllare il funzionamento di allarmi antifurto, porte automatiche, telefoni, trasmettenti ed apparecchi similari.	
Vic switch (4012)	225.000
Possono essere collegati fino a 16 VIC 20 con un floppy e una stampante (distanza massima 1500 mt.).	
Interfaccia IEEE 488 (T-1)	175.000
Interfaccia centronics (T-3)	115.000
RS232-C adapter (1011-A)	75.000
RS132-C adapter (1011-B)	75.000

Commodore 64

CPU 64K RAM (CBM64)	625.000
C 64 Executive (SX 64)	2.350.000
Sistema operativo CP/M (CP/M)	125.000
Consente di programmare il Commodore 64 in linguaggio CP/M, il più utilizzato sui Personal Computers. Permette inoltre di accedere alla enorme libreria di Software applicativi CP/M.	
Pet speed (6411)	95.000
Compilatore basic che aumenta la velocità di esecuzione dei programmi di circa 40 volte.	

Accessori per Vic e Commodore 64

Stampante plotter a colori (1520)	375.000
80 caratteri, per linea, 4 colori, alla risoluzione di 0,2 mm per passo.	
Unità stampante (MPS 801)	450.000
Stampa velocemente su carta normale quanto appare sul video: programmi, lettere, dati, grafici.	
Unità stampante (1526)	595.000
Stampante 80 colonne, bidirezionale, 60 CPS, spaziatura programmabile, trazione a frizione o a trattore.	
Registratore dedicato (1530)	120.000
Per memorizzare facilmente programmi e dati su normali cassette magnetiche.	
Floppy disk drive (1541)	585.000
Veloce unità di memoria di massa ad alta capacità. Può immagazzinare fino a 170.000 caratteri su ogni singolo disco.	
Monitor monocromatico (1601)	285.000
A fosfori verdi 12".	

Per giocare

Comando per giochi (Joystick) (1311)	13.500
Permette di muoversi in tutte le direzioni, di iniziare i vari giochi di movimento e di "sparare".	
Comando a manopola per giochi (Paddle) (1312)	22.500
Adatto per i giochi a 2 persone, esegue movimenti in orizzontale e verticale.	

Commodore 4000

CPU 16K RAM (CBM 4016)	1.285.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.	
CPU 32K RAM (CBM 4032)	1.385.000
18K ROM, BASIC 4.0 residente, video 40 colonne per 25 righe, tastiera semigrafica.	

Commodore 9000

Doppia CPU 134K RAM (CBM 9000)	2.350.000
Micro Main Frame Computer a doppia CPU (6502 - 6809) compatibile con tutte le periferiche Commodore della serie 8000. Include 5 linguaggi di programmazione. (COBOL, FORTRAN, TCL PASCAL, UCSD PASCAL, APL).	

Commodore 8000

CPU 32K RAM (8032 SK)	1.675.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata.	
CPU 96K RAM (8096 SK)	2.150.000
18K ROM, Basic 4.0 residente, video orientabile e basculante 80 colonne per 25 righe, tastiera commerciale separata. Include sistema operativo PM/96.	

Commodore 600

Indicato per applicazioni industriali, collegamento a strumentazione, controllo numerico, ecc. Utilizza monitor in commercio.	
CPU 128K RAM (610)	2.150.000
CPU 128K RAM espandibile internamente a 256K e esternamente a 960K, interfaccia RS232C, IEEE 488, porta utente a 8 Bit. Compatibile con tutte le periferiche Commodore della serie professionale.	
CPU 256K RAM (620)	2.550.000
CPU 256K RAM espandibili esternamente a 960K. Caratteristiche uguali al Mod. 610.	
Monitor (1601)	285.000
Monocromatico a fosfori verdi, 12".	

Commodore 700

CPU 128K RAM (710)	2.850.000
CPU 128K RAM espandibili internamente a 256K ed esternamente a 960K. Video orientabile e basculante 80 colonne per 25 righe. Compatibile con tutte le periferiche Commodore delle serie professionali.	

CPU 256K RAM (720)	3.250.000
Monitor a colori 14" con audio. (1701)	430.000

Dischi

Floppy disk drive (2031)	650.000
Unità di memoria di massa ad alta velocità. Capacità 170KB. Drive singolo.	
Floppy disk drive (4040)	1.900.000
Unità di memoria di massa ad alta velocità. Capacità 343KB. Drive doppio.	
Floppy disk drive (8050)	2.125.000
Drive doppio 1M byte in linea.	
Floppy disk drive (8250 H.P.)	2.450.000
Drive doppio, doppia faccia, doppia densità 2M byte in linea.	
Floppy disk drive (8250 L.P.)	2.600.000
Drive doppio, doppia faccia, doppia densità, 2M byte in linea.	
Hard disk (9060)	6.200.000
Tecnologia Winchester, .5M byte in linea.	
Hard disk (9090)	6.700.000
Tecnologia Winchester, 7.5M byte in linea.	

Stampanti

Stampante (4023 P)	625.000
Bidirezionale ad aghi, 60 CPS, 80 colonne.	
Stampante (MPP 1)	1.195.000
Stampante ad aghi 150 CPS, 132 colonne, bidirezionale, trascinamento a trattore.	
Stampante (6400)	2.950.000

Stampante a margherita, 40 CPS, 136 colonne passo pica, 163 colonne passo élite, bidirezionale, utilizzabile anche con carta da bollo, trascinamento a frizione o a trattore.

Accessori.

Microprocessore 32K RAM (MUPET II)	2.500.000
Per connettere, in rete fino a 16 CPU RS232, IEEE 488, centronics. Il prezzo include (configurazione minima): controller, terminator, 3 moduli, cavi, cavo IEEE/PET (per la versione SK).	
Singolo modulo aggiuntivo:	325.000
1 modulo	
1 cavo 6 piedi.	
Nuovo sistema operativo (PM 96)	95.000
Per 8096SK o per 8032SK con B - 1 oppure con B - 2. Può gestire fino a 16 programmi residenti simultaneamente in memoria. Da a disposizione 26K per le variabili e 53K per i programmi. Potenza inoltre il Basic con altri comandi.	
64K RAM (B-1)	575.000
Scheda di ampliamento memoria per 8032 e nuovo sistema operativo "PM 96".	
CP/Maker (B-2)	1.450.000
Incrementa la memoria interna di 64K RAM e permette l'uso di tutti i programmi CP/M. 8 bit disponibili. Compatibile con la serie 3000/4000/8000.	
Scheda ad alta risoluzione grafica (B-3)	720.000
Compatibile ai sistemi della serie 8000.	
Cavo PET/IEEE 488 (C-1)	85.000
Cavo IEEE 488/IEEE (C-2)	95.000
Accoppiatore acustico (8010)	595.000
300 baud/sec.	

"Commodore Shop" MILANO

massima professionalità su tutti i prodotti
2 punti vendita in città



L'UFFICIO 2000 - Via Ripamonti, 213 - Milano
Tel. 5696570 /5696573

SOFTWARE

Una ricca e selezionata biblioteca di programmi:
contabilità aziendale, gestione negozi, fatturazione, magazzino, ventilazione iva, allegati, gestione archivi dati, ord processing, mailing, list, tutto sull'energia civile ecc... ecc... ecc...

HARDWARE

Riparazioni
Applicazioni speciali
Permute & occasioni

!! ATTENZIONE !! TOTOCALCIO SYSTEM

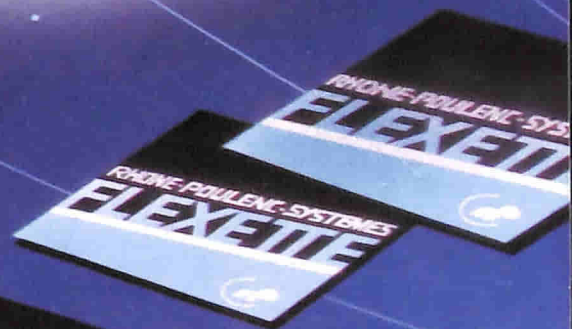
Un programma di altissimo livello, completamente automatico e semplice da usare pr un quasi sicuro 13

"THE LAST ONE"

Un programma che permette anche ai meno esperti di costruire procedure complesse

FLEXETTE

viaggio nella
perfezione



seguite le vostre guide:

**RHÔNE
POULENC
SYSTEMES**
settore informatica
concessionari autorizzati

TECNODATA s.a.s.
di Rossolini Mauro & Dall'Olio Attilio
Via Mazzini 12 (gall. superiore)
43100 PARMIA
Tel. 0521 / 25 079

PROGRAMMA UFFICIO s.a.s.
di Ferrero Enrica & C.
Corso Francia 92/A
10093 COLLECNO (Torino)
Tel. 011 / 41 13 565

SDC di Brignoli Giuseppe & C. s.a.s.
Largo Promessi Sposi 5
20142 MILANO
Tel. 84 35 593 / 84 66 538

DATAPLAN s.a.s.
Via Casa di Risparmio 9
39100 BOLZANO
Tel. 0471 / 47 721

MIDA s.r.l.
Via Diotro Filippini 1/A
37121 VERONA
Tel. 045 / 59 05 05

BRENUANI MASSIMO
Via Peccioli 30 (uff. via Chiusi 76)
00139 ROMA
Tel. 06 / 81 27 665

CSS s.n.c. di Fornasaro A. & G.
Via Fra P. Sarpi 8/A
50136 FIRENZE
Tel. 055 / 67 96 30

TESIN & C. s.r.l.
Via Caravaggio 82
80126 NAPOLI
Tel. 081 / 64 31 22 - 64 67 52

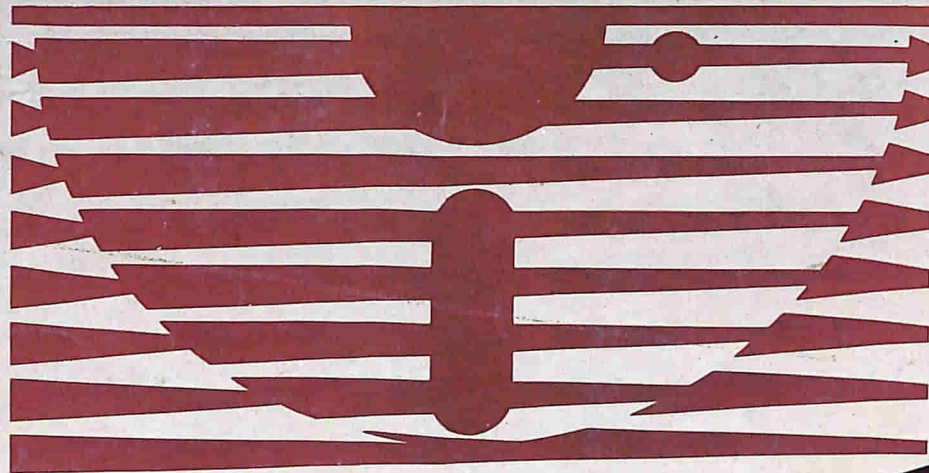
GESCOM s.n.c.
Via Resuttana, 358
90146 PALERMO
Tel. 091 / 518621

STUDIO SINTESI
Via Aldighieri 61
44100 FERRARA
Tel. 0532 / 32618

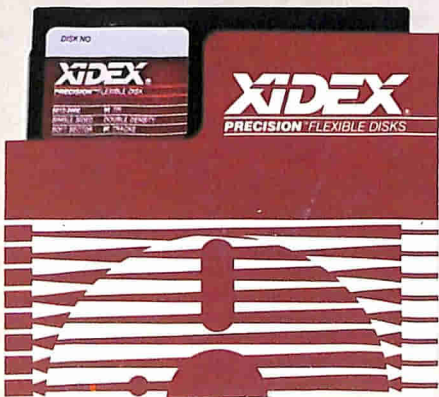
memorie magnetiche per computer.

Ricerca
e Tecnologia

XIDEX



presentano il primo dischetto
di precisione



- Formulazione ossido ad alta densità lineare
- Coercitività fino a oltre 600 oersted
- Clipping level al 65%
- Densità fino a 18000 B P I
- Capacità fino a 5 MB
- Disponibili i nuovi dischetti da 3,5"



 **HC MIL** srl
hc magnetic International line

via passeroni 6 - 20135 milano
tel (02) 577477 - 598353 - telex 340216

 international
data products s.r.l.

corso di porta nuova 34 - 20121 milano
tel. (02) 661491 - 667841