

## Merge

di Aurelio Tontini - Roma

"Caro Leo, non credo di chiedere troppo, ma gradirei esser citato in modo esplicito quando pubblicate miei programmi. Ti saluto, Aurelio".

Chi scrive è Aurelio Tontini, l'autore dell'articolo RENUM-DELETE pubblicato nel software VIC del numero 17 di MCmicrocomputer.

Il pover'uomo, dopo notti insonni passate a redigere i programmi nonché il pezzo (che è tutto suo) si vede confinato (per una nostra dimenticanza) ad un A. Tontini apparso in testa ai listati... è un po' poco, nonostante tutto.

Pubblichiamo doverosa correzione: il lavoro apparso sul n° 17 è tutto suo, come pure suo è il MERGE che andiamo a presentare in questo numero.

## Retrosceca

Il merge serve ad unire due programmi, aggiungendo le linee di differente numero tra i due e, nel caso di un identico numero di linea, pone nel programma complessivo solo quella del secondo listato. Questo lavoro fa parte d'un trittico del Tontini, ma una curiosa anomalia ci impedì di pubblicarlo con gli altri, rendendo necessario interpellare l'autore: la cosa ebbe buoni sviluppi, tanto che gli dedichiamo un capoverso dell'articolo.

L'anomalia consisteva nell'apparizione di una linea fantasma 17703 RA UN \*, che poi veniva ribattuta (con il trucco illustrato nel numero citato, ma che ricorderemo più avanti) dando luogo ad un totale blocco del sistema, con la scomparsa dello schermo e la messa in funzione di uno degli oscillatori del 6561.

Il problema era: da dove cavolo viene questo affare?

Qualche telefonata all'amico Aurelio, ed ecco spiegato l'arcano.

Il VIC salva su nastro una zona intera: questa corrisponde al programma solo se nessuno, prima o dopo il caricamento, gioca con i puntatori, cosa che invece ci era capitata.

Il programma di salvataggio, dunque, vedeva qualcosa dopo il triplo zero di fine programma, e lo registrava; la successiva esecuzione dava luogo alle stranezze di cui sopra. Tanto per cambiare, il lavoro dell'inneffabile Tontini era perfetto.

## Premesse al merge

Riassumiamo in questa sede alcune delle cose già dette, a proposito di queste t e

Programma  
MERGE

```

1 P$=" PRIMO":GOSUB60:GOTO3
2 P$=" SECONDO":F=1:GOSUB60
3 UL=PEEK(55)+PEEK(56)*256-2:H=INT(UL/256):L=UL-H*256:POKE51,L:POKE55,L:POKE52,H
:POKE56,H
4 GETA$:IFA$=""THEN4
5 IFASC(A$)=133ANDF=0THEN10
6 IFASC(A$)=133THEN40
7 GOTO4
10 GOSUB50:PRINT"XXXXXXXXXXXX#63999REM":PRINT:PRINT"PG43,1:PG44,18:R/256":END
40 GOSUB50:PRINT"XXXXXXXXXXXX#PG43,1:PG44,18:R/800":END
50 LH=UL+1:LL=UL:LO=PEEK(45)+PEEK(46)*256-2:H=INT(LO/256):L=LO-H*256
51 POKELH,H:POKELL,L
52 POKE198,10:FORI=0TO5:POKE631+I,13:NEXT
53 PRINT"PG43,";L;"PG44,";H;"LG":RETURN
60 PRINT"CARICAMENTO":PRINTP$;"PROGRAMMA":PRINT"PREPARA LA CASSETTA":PRINT
T"POI PREMI 'F1':RETURN
80 UL=PEEK(55)+PEEK(56)*256-5:A=INT(UL/256):B=UL-A*256:POKE51,B:POKE55,B:POKE52,
A:POKE56,A
90 POKEUL+3,PEEK(45):POKEUL+4,PEEK(46):GOTO100
95 FB=PEEK(55)+PEEK(56)*256+5:POKEFB-2,PEEK(45):POKEFB-1,PEEK(46)
100 DIMC$(75):FORI=0TO75:READC$(I):NEXT:PRINT"FB=PEEK(55)+PEEK(56)*256+5
110 POKE198,10:FORI=0TO4:POKE631+I,13:NEXT
120 PRINT"PG43,";PEEK(FB+2);"PG44,";PEEK(FB+3);"":PRINT:L=PEEK(FB)+PEEK(FB+1)
#256
130 HE=PEEK(L+3):LE=PEEK(L+2):POKEFB-3,HE:POKEFB-4,LE:ET=LE+HE*256:PRINT:PRINTET
;"#";
140 FORK=4TO999:P=PEEK(L+K):IFP=0THEN200
150 IFP=34THENGOSUB500
160 IFP>127THEN180
175 PRINTCHR$(P);:NEXT
180 IFP=255THENPRINT" ";:NEXT
182 IFF=1THEN175
185 C$=C$(P-128):IFRIGHT$(C$,1)="#"THENC$=LEFT$(C$,LEN(C$)-2)+CHR$(ASC(MID$(C$,L
EN(C$)-1,1))+128)
190 PRINTC$;:NEXT
200 L=L+K+2:K=999:NEXT:POKEFB-5,PEEK(L)
210 PRINT"PG43,1:PG44,18:R/300":END
300 PRINT"FB=PEEK(55)+PEEK(56)*256+5
320 ET=PEEK(FB-4)+PEEK(FB-3)*256:CP=PEEK(FB-2)+PEEK(FB-1)*256
330 SC=PEEK(45)+PEEK(46)*256-CP:I2=PEEK(FB)+PEEK(FB+1)*256+SC:H2=INT(I2/256):L2=
I2-H2*256
340 POKEFB,L2:POKEFB+1,H2:POKE198,10:FORI=0TO4:POKE631+I,13:NEXT
350 PRINT"PG43,";L2;"PG44,";H2;"":PRINT:PRINT:PRINT"PG43,1:PG44,18:"
;
370 IFPEEK(FB-5)=0THENPRINT"R/390":END
380 PRINT"R/95":END
390 I=PEEK(55)+PEEK(56)*256+7:H=INT(I/256):L=I-H*256:POKE43,PEEK(I):POKE44,PEEK(
I+1)
399 UB=PEEK(45)+PEEK(46)*256:POKEUB-8,0:POKEUB-7,0:PRINT" ":END
400 DATAEN#,FO#,NE#,DA#,IN#,INPUT,DI#,RE#,LE#,GO#,RU#,IF,RES#,GOS#,RET#,REM,ST#,
ON,WA#
401 DATALO#,SA#,VE#,DE#,PO#,PR#,?,CO#,LI#,CL#,CM#,SY#,OP#,CLO#,GE#,NEW,TA#,TO,FN
,SP#
402 DATATH#,NOT,STE#,+,-,*,/,^,AN#,OR,>,<,SG#,INT,AB#,US#,FR#,POS,SG#,RN#,LOG,
EX#
403 DATACOS,SI#,TAN,AT#,PE#,LEN,STR#,VA#,AS#,CH#,LEF#,RI#,MI#,GO
500 IFF=0THENF=1:RETURN
510 F=0:RETURN
    
```

Listato del programma di MERGE. Facciamo risalire ancora una volta l'alto livello didattico: notare in linee 400-403 i DATA contenenti le abbreviazioni di tutte le parole riservate del BASIC. Queste in generale vengono riconosciute dall'interprete tramite le due prime lettere seguite dal simbolo #.

routine, insieme al RENUMBER e al DELETE.

Lo scopo principale è sempre mostrare alcuni trucchetti implementabili sul VIC, nonché diverse peculiarità del suo funzionamento (come sistema operativo e come BASIC) senza dover ricorrere a spiegazioni tecniche: il mezzo usato è quello di sfruttare le nozioni in un programma, tra l'altro utile, raggiungendo così un duplice scopo.

Diciamo quindi in modo esplicito che non si tratta del modo più breve per raggiungere il fittizio obiettivo di rinumerare un programma, o di attaccarne un paio.

I trucchi-base del trittico del Tontini sono essenzialmente due: la simulazione e della battuta d'una istruzione, più l'uso del bianco per mimetizzare con lo sfondo le istruzioni dirette.

Vediamoli entrambi.

(Riprendiamo dall'ormai famoso numero 17, pg. 79).

È possibile simulare la digitazione da tastiera.

Durante l'esecuzione d'un programma il buffer di tastiera è inaccessibile (tranne che con apposite istruzioni tipo INPUT o GET).

È però possibile forzarlo agendo sul puntatore allocato in 198, con una POKE 198, Y (Y è il numero di caratteri immagazzinabili, al massimo 10), e specificando il carattere desiderato nelle locazioni da 631 a 640 con altre POKE T,U (T è la locazione da 631 a 640, mentre U è il codice ASCII del carattere di cui vogliamo simulare la digitazione).

Quando il calcolatore restituisce all'operatore l'uso della tastiera, cosa che avviene alla fine del programma oppure con STOP o END, gli Y caratteri annidati nelle locazioni T saltano fuori visualizzandosi sullo schermo, eventualmente realizzando comandi diretti se sono tali i caratteri dati.

Se allora all'esecuzione dell'END (inteso come istruzione) faremo capitare il cursore su una istruzione scritta sullo schermo da un PRINT precedente, avendo immagazzinato nella locazione 631 un valore 13 (= RETURN), l'EDITOR di schermo si comporterà come se qualcuno avesse digitato quella istruzione e poi battuto il RETURN.

La seconda cosa è assai più semplice: queste istruzioni dirette, essenziali per il funzionamento del programma, potrebbero dar fastidio sullo schermo soprattutto a coloro che, non preoccupandosi del sistema usato, vogliono solo usare il programma.

La trovata, in questo caso, è semplice: basta non dare il colore all'istruzione, che risulta quindi a noi invisibile (ma non per questo meno valida per il VIC).

## Il programma

Gira con almeno 7K di RAM (quindi con espansioni da 8K in su). La scelta ci pare appropriata, dato che il VIC inespanso, tolto circa 1K per il MERGE, avrebbe solo un paio di kilobyte per gli altri due programmi.

La prima cosa da fare è leggere da nastro il primo programma, le cui istruzioni in caso di conflitto per il numero di linea verranno cancellate e sostituite con quelle del secondo programma.

A questo punto se muoviamo i puntatori che dicono al BASIC dov'è il programma appena letto, ponendoli alla fine di questo, l'interprete vedrà una zona di memoria vuota su cui poter tranquillamente copiare il secondo programma.

Viene ora il bello. Ponendo i puntatori nel corretto modo, l'interprete vedrà entrambi i programmi, e il LIST ve li mostrerà di fila, ognuno con i suoi numeri di linea.

Il programma prende ognuna delle istruzioni del secondo programma, le ribatte nel primo, sposta i puntatori alla fine del primo programma, ribatte il solo numero di linea dell'istruzione appena trasferita nel programma principale (cancellandola dal secondo) e ricomincia. La cancellazione linea per linea delle vecchie istruzioni è dovuta all'eventuale possibilità di finire la memoria, situazione così scongiurata.

## Il listato

Anche qui andremo per sommi capi.

I puntatori utilizzati sono nelle tre copie di locazioni 43-44 (inizio del BASIC), 45-46 (fine del programma) e 55-56 (fine della memoria disponibile); il mezzo con cui questi due byte contengono un numero compreso tra 0 e 65535 (in effetti sul VIC il massimo è 63999) è semplice: basta moltiplicare per 256 il contenuto del secondo byte, ed aggiungere quello del primo. Le altre locazioni interessate sono la 198 per il forzamento del buffer di tastiera, quelle dal 631 al 640 per immagazzinare i codici ASCII citati, e la condizione IF ASC(AS) = 133 che ci dice se è stato premuto F1.

Infine spieghiamo a cosa servono i DATA alfanumerici delle ultime linee.

Ogni linea BASIC, sul VIC, può essere lunga fino a circa 80 caratteri: questo vuol dire, ad esempio, che possiamo mettere fino a 40 punti interrogativi (corrispondenti all'istruzione PRINT) seguiti dai due punti di fine istruzione, e che — comunque sia — usando i comandi abbreviati mostrati in qualunque testo sul VIC (VIC Revea-

led, Reference Programmer's, EVM, V.I.C.?, manuale in dotazione) possiamo compattare molte istruzioni in una sola linea.

Quando queste vanno ad essere mostrate dall'editor di schermo, però, questo usa la forma normale, sicché nell'esempio fatto troveremo una linea di  $6 \cdot 40 = 240$  caratteri!, che in nessun modo verrà poi accettata dall'interprete.

L'unico modo per poter ribattere queste linee è affidarsi al loop di lettura DATA in linea 100, che per ogni comando sostituisce la forma espansa con quella compatta. Dal punto di vista della didattica, tanto di cappello.

## Concludendo

I programmi sono a prova di bomba.

Il movimento che si sta creando intorno al VIC è più che ampio, nonché in velocissima ascesa: prova ne siano i numerosi programmi che riceviamo ogni mese.

La corretta procedura è questa: inviate il listato, delle note al listato (eventualmente l'articolo già pronto per il visto della redazione) e magari il nastro su cui il programma è registrato, specie se il programma è molto lungo.

MC

### INVIAATECI I VOSTRI PROGRAMMI!

Se, qualunque sia la vostra macchina, avete realizzato programmi o routine che ritenete possano interessare altri lettori, inviateceli. Saranno esaminati e, se pubblicati, ricompensati con valutazioni approssimativamente fra le 30 e le 100.000 lire, secondo la complessità, la genialità, l'originalità e la presentazione del materiale e della documentazione (listati, diagrammi, commenti ecc.). Per ragioni organizzative non possiamo impegnarci, salvo eventuali accordi presi prima dell'invio, alla restituzione dei materiali, che resteranno di proprietà della redazione che si impegna a non divulgarli (se non tramite la rivista) senza l'autorizzazione dei rispettivi autori.

# SICOB

# 83



PUBLICIS

X 373

## PARIGI CAPITALE DELL'INFORMATICA

TUTTO IL MONDO DELL'INFORMATICA SARÀ  
A PARIGI IN SETTEMBRE IN OCCASIONE  
DI DUE MANIFESTAZIONI A LIVELLO MONDIALE.

### SICOB 21 - 30 SETT.

SALONE INTERNAZIONALE DELL'INFORMATICA,  
TELEMATICA, COMUNICAZIONE, ORGANIZZAZIONE  
E AUTOMATIZZAZIONE DELL'UFFICIO.

### IFIP 83 19 - 23 SETT.

9° CONGRESSO MONDIALE DELL'INFORMATICA

Per informazioni rivolgersi a: Promosalons Italia Saloni Specializzati Francesi  
Viale Teodorico 19/2 - 20.149 Milano - Tel. (2) 345.86.51/2/3 - Telex 333 448 Prosal 1