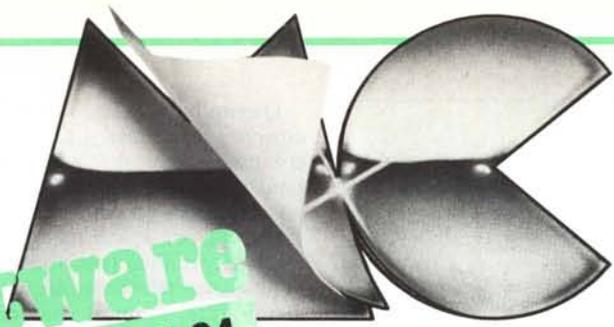


# software COMMODORE 64



Con grande gaudio vi annunciamo la fine di buona parte dei problemi derivanti dai caratteri speciali dei computer Commodore: d'ora in avanti, infatti, useremo una speciale interfaccia che al loro posto mette una sigla tra virgolette, per cui la Q in campo inverso diventa CD (Cursor Down), il cuoricino diventa CS (Clear Screen) e così via. Dato che la cosa si ottiene via hardware, i listati rimangono completamente attendibili, a differenza di quanto accade in altre riviste dove le correzioni vengono apportate manualmente o tramite un programma (sistemi non completamente affidabili).

Pubblichiamo a parte una lista delle equivalenze con i nuovi caratteri speciali, oltre ad una prova dell'interfaccetta Cardco (leggete i reportage dalle mostre? Era nello scorso CES, MC 31).

Il software riguarda un eccellente programma realizzato dal nostro amico Giorgio Leo Rutigliano, apparso su MC più volte

— SuperList, RTTY — del quale continuiamo ad apprezzare i lavori, davvero professionali anche nella documentazione. Stavolta ci propone uno Spreadsheet, ovvero un tabellone elettronico: la cosa eccellente di questo programma è che è scritto interamente in Basic, per di più opportunamente commentato, e così assume un elevatissimo valore didattico. Torniamo al programma: il formato della pagina a disposizione è 26x52 in cui lo schermo è una finestra 4x20. Le particolarità di questo pacchetto sono molteplici, ma lascio ben volentieri la parola all'articolo dell'autore, che è il manuale del programma. Prima, però, una nota: Spreadsheet è disponibile tra le cassette di MC ad un prezzo un po' più elevato del solito, perché queste vengono fornite in un lato del programma in Basic, e nell'altro del suo compilato — con ogni probabilità realizzato con il Petspeed — che risolve molti dei problemi di velocità intrinseci al Basic.

I "tabelloni elettronici" hanno avuto, negli ultimi tempi, un meritissimo successo, Visicalc e similari (Calcstar, Multiplan, VisiOn) presentano l'importantissimo merito di aver avvicinato il computer ad una sfera di utilizzatori non-programmatori e di aver contribuito ad una maggiore diffusione di queste macchine. Ciò perché consentono di ottenere ottimi risultati grazie ad una eccezionale flessibilità operativa unita ad una enorme semplicità di uso. Gli Spreadsheet, infatti, non richiedono conoscenze specifiche di programmazione, necessitano di un periodo di training molto limitato, hanno comandi semplici da ricordare ma estremamente efficienti, impegnano poco l'utilizzatore perché il loro approccio alla soluzione del problema è estremamente naturale. L'unica "nota dolens" della situazione è, purtroppo, il prezzo. Un "tabellone elettronico" costa cifre notevoli, che non ne giustificano l'acquisto a scopo hobbistico.

Proprio per questo motivo un po' di tempo fa un mio amico, affascinato dalle prestazioni del Visicalc, ma leggermente intimorito dal suo prezzo mi chiese: 'Perché non ne fai una versione (magari un tantino ridotta) per il 64?'. Così, un po' per gioco, un po' per realizzare qualcosa di nuovo (o di vecchio?), nacque il programma "SPREADSHEET". Poiché il fine primario del programma risulta essere quello didattico, ho preferito scriverlo interamente in Basic evitando routine assembler anche per le cose più ovvie (come, ad esempio, la gestione del video).

----- TUTTI I CODICI DI CONTROLLO -----				----- CURSORE E SCHERMO -----			
----- ECCO I COLORI CBM -----				CAR.	CODICE	PREMERE	FUNZIONE
CAR.	CODICE	PREMERE	FUNZIONE				
█	{BK}	144	SH+1	NERO			
█	{WH}	5	SH+2	BIANCO			
█	{RD}	28	SH+3	ROSSO 1			
█	{CY}	159	SH+4	CYAN*			
█	{PU}	156	SH+5	PORPORA			
█	{GN}	30	SH+6	VERDE			
█	{BL}	31	SH+7	BLU			
█	{YL}	158	SH+8	GIALLO			
█	{OR}	129	CBM+1	ARANCIO			
█	{BR}	149	CBM+2	MARRONE			
█	{LR}	150	CBM+3	ROSSO 2			
█	{G1}	151	CBM+4	GRIGIO 1			
█	{G2}	152	CBM+5	GRIGIO 2			
█	{LG}	153	CBM+6	VERDE 2			
█	{LB}	154	CBM+7	BLU 2			
█	{GB}	155	CBM+8	GRIGIO 3			
	{HM}	19		HOME		HOME	
	{SC}	147		SH+ HOME		CLR	
	{CR}	29		CUR. 1		CUR. DES	
	{CL}	157		SH+CUR. 1		CUR. SIN	
	{CD}	17		CUR. 2		CUR. GIU'	
	{CU}	145		SH+CUR. 2		CUR. SU	
----- ALTRE COSE UTILI -----				CAR.	CODICE	PREMERE	FUNZIONE
	{F1}	133		F1		F1	
	{F2}	137		SH+F1		F2	
	{F3}	134		F3		F3	
	{F4}	138		SH+F3		F4	
	{F5}	135		F5		F5	
	{F6}	139		SH+F5		F6	
	{F7}	136		F7		F7	
	{F8}	140		SH+F7		F8	
	{RV}	18		CTL+ 9		RV SI	
	{RO}	146		CTL+ 0		RV NO	

Ecco la rappresentazione dei codici di controllo dei nuovi listati di MC: si tratta di un miglioramento senz'altro decisivo.

```

10 REM *****
12 REM SPREADSHEET
20 REM DI GIORGIO LEO RUTIGLIANO
30 REM
60 REM RELEASE 1.2 / 5.84
70 REM
71 REM *****
80 PRINT "(SC)(RV)(CD)(CD):SPREADSHEET* BY GIORGIO RUTIGLIANO 1984":FOR I=1 TO 3
90:GOTO 110
90 GOSUB 4270:GOSUB 2090:GOSUB 2010
100 PRINT "(HM)SPREADSHEET 1.2"TAB(40)"(C) GIORGIO RUTIGLIANO 05.1984(SS)":
110 POKE 2020,32
120 REM ***
130 REM *** ANALIZZATORE COMANDI
140 REM ***
150 GET AS:IF AS="" THEN 150
160 IF AS="(CL)" THEN UX=UX-1:GOSUB 2220
170 IF AS="(CR)" THEN UX=UX+1:GOSUB 2220
180 IF AS="(CU)" THEN UY=UY-1:GOSUB 2220
190 IF AS="(CD)" THEN UY=UY+1:GOSUB 2220
200 IF AS="(HM)" THEN UX=0:UY=0:GOSUB2220
210 IF AS=")" THEN GOSUB 2690
220 IF AS="/" THEN 720
230 IF AS="V" THEN 400
240 IF FNQ(0)=34 THEN 310
250 IF FNQ(0)=13 THEN WX=0:UY=UY+1:GOSUB 2220
260 IF FNQ(0)=141 THEN WX=FX:UY=UY+1:GOSUB 2220
270 IF AS="(SC)" THEN WX=FX:UY=FY:GOSUB 2220
280 IF AS="!" THEN 540
290 GOTO 110
300 REM ***
310 REM *** ROUTINE GESTIONE COMANDI
320 REM ***
330 REM
340 REM +++ INTR.LABEL
350 REM
360 PRINT CL$+ERS"ENTRY LABEL"CES"(HM)":;YS=0
370 GOSUB 2480
380 IF AS<>" " THEN CS(UX,UY)="L"+AS:GOSUB 2400
390 POKE 2020,32:PRINTCL$:;GOTO110
400 REM
410 REM +++ INTR.VALUE
420 REM
430 PRINT CL$+ERS"ENTRY VALUE"CES"(HM)":;YS=0
440 GOSUB 2480:IF AS=" " THEN POKE 2020,32:GOTO110
450 GOSUB 2850:CS(UX,UY)="V"+AS:IF ER% THEN GOTO110
460 IF MY<UY THEN MY=UY
470 IF MX<UX THEN MX=UX
480 IF MAX=0 THEN 540
490 V(UX,UY)=OP(0):GOSUB 2400
500 POKE 2020,32:PRINTCL$:;GOTO110
510 REM
520 REM +++ RICALCOLD SHEET
530 REM
540 PRINT CL$:;POKE2020,32
550 IF RIX=0 THEN FOR CX=0 TO MX
560 FORCY=0TOMY:IFRIN%THENFORCX=0TOMX
570 OP(0)=0:IFASC(CS(CX,CY)+"(SS)")<>96 THEN 660
580 AS=MID$(CS(CX,CY),2):GOSUB 2850:V(CX,CY)=OP(0):IF ER%THEN V(CX,CY)=1E33
590 IF CX<FX OR CX>FX+3 THEN 660
600 IF CY<FY OR CY>FY+3 THEN 660
610 AS=MID$(STR$(V(CX,CY)),2+(V(CX,CY)<0),9)
620 PRINT "(HM)(CD)":FOR YS=1 TO (CY-FY+1):PRINT "(CD)":;NEXT:PRINT "(CR)(CR)(CR)";
630 IF (CX-FX)>0 THEN FOR XS=1 TO CX-FX:PRINT "(CR)(CR)(CR)(CR)(CR)(CR)(CR)(CR)(CR)":;NEXT
640 IF CX=UX AND CY=UY THEN PRINT "(RV)":
650 PRINT LEFT$(BL$,9-LEN(AS))AS"(RO)":
660 IF RIX=0 THEN NEXTCY
670 NEXTCX:IF RIX THEN NEXTCY
680 POKE2020,32:GOTO110
690 REM
700 REM +++ ACK /
710 REM
720 PRINT CL$+ERS"COMANDI: BCEGFS":;POKE 2020,32
730 GET AS:IF AS="" THEN 730
740 PRINTCL$:;IF AS="B"THENPRINTCL$CES:;AS="":GOTO 380
750 IF AS<>"E" THEN 890
760 REM
770 REM --- /E
780 REM
790 YS=LEN(CS(UX,UY))-1
800 IF PEEK(2019)=12 THEN PRINT ER$"EDIT LABEL":;GOTO 370
810 IF PEEK(2019)=22 THEN PRINT ER$"EDIT VALUE":;GOTO 440
820 GOTO 110
830 IF AS<>"C" THEN 910
840 REM
850 REM --- /C
860 REM
870 PRINTCL$+ERS"CLER:CONFERMA(S/N)?"
880 GET AS:IF AS="" THEN 880
890 PRINT CL$:;IF AS<>"S" THEN 110
900 CLR:PRINT"(SC)":;GOTO 90
910 IF AS<>"S" THEN 1320
920 REM
930 REM --- /S
940 REM
950 PRINT CL$+ERS"SAVE: L S Q":
960 GET AS:IF AS="" THEN 960
970 PRINT CL$:;IF AS<>"L" THEN 1250
980 REM
990 REM --- /SL
1000 REM

```

Questo ha leggermente rallentato l'esecuzione, ma contribuisce fortemente alla comprensibilità del programma stesso. Uno spreadsheet, infatti, è un insieme estremamente complesso.

Per chi non lo conosce per niente dirò che il programma trasforma il calcolatore in un grosso tabellone, di cui lo schermo è una finestra, composto da una serie di caselle, individuate da una lettera (la colonna) e due cifre (la riga). In ognuna di queste caselle è possibile inserire tre distinti tipi di dati: dati alfabetici, ovvero etichette utili solo a rendere più comprensibili i dati contenuti nel tabellone, numeri, formule. Questi ultimi due elementi sono la forza principale del sistema di calcolo. Una formula può richiamare il contenuto di altre caselle. Ad esempio, se io volessi addizionare il contenuto della casella B23 a quello della casella A12 e metterle il contenuto nella cella D33, dovrei mettere nella cella D33 la formula "B23 + A12". A questo punto, stabilita la correlazione tra due caselle, mi è sufficiente modificare il valore del numero (o della formula) contenuta in una delle due caselle B23 o A12 per vedere cambiare anche quella della cella D33.

Questo esempio, anche se estremamente semplice, consente di rendersi conto della estrema potenza del mezzo Spreadsheet. Moltissimi problemi, infatti, trovano soluzione naturale e semplice, senza richiedere sforzi di programmazione. Bilanci familiari, calcoli condominiali, previsioni di spesa e altro, senza limiti alla fantasia.

Lo spreadsheet, inoltre, implementa alcuni altri comandi accessori per semplificarne l'uso. Ad esempio la possibilità di posizionare il cursore su una cella qualsiasi, di stampare il tabellone, di conservare e richiamare i dati, etc. In Spreadsheet la struttura dei comandi ricorda molto quella del Visicalc, con alcune limitazioni dovute al tipo di impostazione scelta.

## Il programma

Il programma è logicamente suddiviso in vari blocchi:

- analizzatore di comandi (linee 120-1970): rappresenta il corpo principale del programma. Scopo è quello di interpretare correttamente i comandi, rifiutarli se errati e trasferire il comando alla routine che esegue le elaborazioni connesse al comando stesso;

- gestione video (linee 2000-2840), che esegue tutte le operazioni sulla finestra video: scrolling, aggiornamento del display e delle celle dopo il calcolo, posizionamento del cursore;

- elaborazione calcoli (linee 2880-4040), che effettua tutti i calcoli connessi alla gestione del tabellone, gestisce stack e funzioni complesse, ed effettua una diagnostica sulle formule impostate;

- gestione errori (linee 4050-4260), che comunica eventuali errori all'operatore so-

```

1010 PRINT CL$ERS:"LOAD:CONFERMA (S/N) ?":
1020 GET AS:IF AS="" THEN 1020
1030 PRINT CL$:IF AS<>"S" THEN 110
1040 IF CA% THEN PRINT "(SC)"
1050 POKE1944,CA%:CLR:GOSUB 4270:CA%=PEEK(1944)
1060 IF CA% THEN PRINT "(HM)":OPEN 1,1,0:OPEN 15,0:GOTO 1140
1070 PRINT CL$ERS:"LOAD:NOME DEL FILE?":CE%:YS=0:GOSUB 2480
1080 IF YS=0 THEN PRINT CL$:CE%:GOTO110
1090 AS=LEFT$(AS,13)
1100 OPEN15,8,15:OPEN1,8,2,AS+".VC,R,S"
1110 INPUT#15,D,D$:IF D=0 THEN 1140
1120 PRINT CL$ERS:"DE"D":D$:
1130 CLOSE 1:CLOSE 15:GOTO110
1140 PRINT CL$ERS:"LEGGO I DATI":
1150 INPUT#1,UX,WY,FX,FY,MAX,R1%:POKE 2020,42
1160 INPUT#1,XS:POKE 2020,32:IF XS<0 THEN 1230
1170 INPUT#1,YS,V(XS,YS),AX:POKE2020,42
1180 FOR I=1 TO AX:INPUT#1,F%:CS(XS,YS)=CF(XS,YS)+CHR$(F%):NEXT:POKE2020,32
1190 IF ASC(CS(XS,YS))<86 THEN 1160
1200 IF MX<XS THEN MX=XS
1210 IF MY<YS THEN MY=YS
1220 GOTO 1160
1230 CLOSE1:CLOSE15:GOSUB2090
1240 PRINT "(HM)"MID$(CS(UX,WY)+BL$+BL$,2,80)CL$:GOSUB2250:GOTO110
1250 IF AS<>"Q" THEN 1330
1260 REM
1270 REM --- /SQ
1280 REM
1290 PRINT CL$ERS:"QUIT:CONFERMA (S/N) ?":
1300 GET AS:IF AS="" THEN 1300
1310 PRINT CL$:IF AS<>"S" THEN 110
1320 PRINT"(SC)(CD)(CD):FINE DEL PROGRAMMA":END
1330 IF AS<>"S" THEN 110
1340 REM
1350 REM --- /SS
1360 REM
1370 IF CA% THEN PRINT "(HM)":OPEN 1,1,1:OPEN 15,0:GOTO 1450
1380 PRINT CL$ERS:"SAVE:NOME DEL FILE?":CE%:YS=0:GOSUB 2480
1390 IF YS=0 THEN PRINT CL$:CE%:GOTO110
1400 AS=LEFT$(AS,13)
1410 OPEN15,8,15:OPEN1,8,2,AS+".VC,U,S"
1420 INPUT#15,D,D$:IF D=0 THEN 1450
1430 PRINT CL$ERS:"DE"D":D$:
1440 CLOSE 1:CLOSE 15:GOTO110
1450 PRINT CL$ERS:"SALVO I DATI":PRINT#1,UX;R$;WY;R$;FX;R$;FY;R$;MAX;R$;R1%
1460 FOR CX=0 TO 25:FOR CY=0 TO 31
1470 AX=LEN(CS(CX,CY)):IFAX=0 THEN 1500
1480 PRINT#1,CX;R$;CY;R$:V(CX,CY);R$:AX
1490 FOR I=1 TO AX:PRINT#1,ASC(MID$(CS(CX,CY),I)):NEXT I
1500 NEXT CY:NEXT CX:PRINT#1,-1:CLOSE 1:CLOSE 15
1510 PRINT "(HM)"MID$(CS(UX,WY)+BL$+BL$,2,80)CL$:GOTO 110
1520 IF AS<>"G" THEN 1650
1530 REM
1540 REM --- /G
1550 REM
1560 PRINT CL$ERS:"GLOBALE: A C D M N R":
1570 GET AS:IF AS="" THEN 1570
1580 PRINT CL$:IF AS="A" THEN PRINT ER$:"CALCOLO AUTOMATICO":MAX=0:POKE2015,32
1590 IF AS="M" THEN PRINT ER$:"CALCOLO MANUALE":MAX=1:POKE2015,13
1600 IF AS="C" THEN PRINT ER$:"CALCOLO PER COLONNE":R1%=0:POKE 2018,8
1610 IF AS="R" THEN PRINT ER$:"CALCOLO PER RIGHE":R1%=1:POKE 2018,18
1620 IF AS="D" THEN PRINT ER$:"SUPPORTO:DISCO":CA%=0
1630 IF AS="N" THEN PRINT ER$:"SUPPORTO:NASTR0":CA%=1
1640 GOTO 110
1650 IF AS<>"P" THEN 1960
1660 REM
1670 REM --- /P
1680 REM
1690 PRINT CL$ERS:"STAMPA FINO A":
1700 GET AS:IF AS="" THEN 1700
1710 A=ASC(AS):IF A=13 THEN 1950
1720 IF A<65 OR A>90 THEN 1940
1730 PRINT AS,"":XS=A-65
1740 GET AS:IF AS="" THEN 1740
1750 A=ASC(AS):IF A=13 THEN 1950
1760 IF A<48 OR A>57 THEN 1940
1770 PRINT AS":YS=(A-48)*10
1780 GET AS:IF AS="" THEN 1780
1790 A=ASC(AS):IF A=13 THEN 1950
1800 IF A<48 OR A>57 THEN 1940
1810 PRINT AS":YS=A-49+YS
1820 IF XS<UX THEN PRINT CL$ERS+"ERRORE IN X":GOTO 110
1830 IF YS<VX THEN PRINT CL$ERS+"ERRORE IN Y":GOTO 110
1840 OPEN 4,4
1850 FOR I=UX TO XS:PRINT#4,CHR$(I+65) "":NEXT:PRINT#4," "
1860 FOR J=UY TO YS:AS=MID$(STR$(J+1),2):IF LEN(AS)<2 THEN AS=" "+AS
1870 PRINT#4,CHR$(RO) "":FOR I=UX TO XS:AS=CS(I,J)+
1880 IF ASC(AS)=86 THEN AS="(SS)"MID$(STR$(V(I,J)),2)
1890 AS=MID$(AS,2,9):AS=LEFT$( " ",9-LEN(AS))+AS
1900 PRINT#4,AS":NEXT:PRINT#4:NEXT
1910 POKE2018,31:IFR1% THEN POKE2018,18
1920 POKE2015,32:IFR1% THEN POKE2015,13
1930 POKE 2016, FN F(0):POKE 2017,48:CLOSE 4:PRINT CL$ERS:GOTO 110
1940 AS="":PRINT CL$ERS+"ERRORE IN GOTO":RETURN
1950 AS="":PRINT CL$ERS:RETURN
1960 GOTO 110
1970 END
1980 REM ***
1990 REM *** SUBROUTINE GESTIONE VIDEO
2000 REM ***
2010 REM
2020 REM +++ AGGIORNA POINTER
2030 REM

```

(continua a pag. 134)

spendendo l'elaborazione del tabellone — inizializzazione (linee 4300-4430), che effettua le operazioni di assegnazione iniziale di valori pre-definiti ad alcune variabili.

## 1. Il display

Questo programma trasforma il display del calcolatore in una finestra mobile su di un tabellone elettronico che può essere scritto, letto e modificato.

### FORMATO DEL DISPLAY

linea editing 1				
linea editing 2	A	B	C	D
1				
2				
	TABELLONE			
19				
20				
linea messaggi 1				
linea messaggi 2				STATUS

Come è possibile vedere il tabellone è organizzato in righe (numerata da 1 a 52) e in colonne (26, denominate A..Z). Lo schermo è una finestra sul tabellone larga 4 caselle e alta 20. Le celle sono denominate da una lettera riferita alla 1ª colonna e da due cifre indicanti la riga. La prima in alto diventa quindi A01, l'ultima in basso Z52. Lo zero è significativo: A1 non viene infatti accettato. In ogni cella è possibile scrivere un titolo, una cifra o una formula. La cella in cui si trova il cursore viene evidenziata in reverse. Le due linee in alto (linea editing 1 e 2) servono per passare dati al programma. La linea in basso, invece, viene utilizzata dal programma per comunicare con l'operatore. Nella parte sinistra appaiono richieste e messaggi di errore. Nella parte destra invece appaiono degli indicatori che permettono all'operatore di conoscere immediatamente alcuni parametri del programma:

STATUS:

M 90 C V ! A23

calcolo man. |  
 memoria libera |  
 ordine di calcolo |  
 posizione cursore |  
 indicatore lavoro |  
 tipo di cella

calcolo manuale: appare quando viene abilitato il calcolo manuale (comando /MM)  
 memoria libera: indicazione percentuale di memoria ancora disponibile per i dati del tabellone.

ordine calcolo: C colonne (default), R righe

tipo di cella: l etichetta, V formula

tipo di lavoro: questo carattere indica che il programma effettua un lavoro e non può accettare comandi. \*# indica movimento dati e gestione display. ! indica fase di calcolo. \*\* indica save/load dati da memoria di massa

(segue da pag. 133)

```
2040 PRINT "(HM)*MID$(C$(UX,UY)+BL$+BL$,2,80);
2050 YS=UY+1;POKE2021,UX+1;POKE2022,INT(Y5/10)+48;POKE2023,YS-INT(Y5/10)*10+48
2060 XS=1147+OX*9+OY*40;POKE 2019,FNSC(ASC(C$(UX,UY)+ " "));
2070 FOR XS=XS TO XS+8:POKEXS,PEEK(XS)AND127:NEXT:XS=1147+(UX-FX)*9+(UY-FY)*40
2080 FOR XS=XS TO XS+8:POKE XS,PEEK(XS)OR128:NEXT:OX=(UX-FX);OY=(UY-FY);RETURN
2090 REM
2100 REM +++ REFRESH DISPLAY
2110 REM
2120 PRINT C$;
2130 FOR XS=FX TO FX+3:PRINT "(RV) "CHR$(XS+65)";:NEXT:PRINT " ";
2140 FOR YS=FY+1 TO FY+20:AS=MID$(STR$(YS),2);IF LEN(AS)<2 THEN AS=" "+AS;
2150 PRINT "(RV)"AS"(RD)";:FOR XS=FX TO FX+3:AS=C$(XS,YS-1)+ " ";
2160 IF ASC(AS)=88 THEN AS="(88)"MID$(STR$(V(XS,YS-1)),2+(V(XS,YS-1)<0))
2170 AS=MID$(AS,2,9):AS=LEFT$( " ",9-LEN(AS))+AS
2180 PRINT AS;:NEXT:PRINT:NEXT
2190 POKE2018,3;IFR1%THENPOKE2018,18
2200 POKE2015,32;IFR1%THENPOKE2015,13
2210 POKE 2016,FNF(0):POKE 2017,48;RETURN
2220 REM
2230 REM +++ POSIZIONAMENTO CURSORE
2240 REM
2250 PRINT CL$;DF=0;POKE 2020,35
2260 IF WX<0 THEN WX=0
2270 IF WY<0 THEN WY=0
2280 IF WX>25 THEN WX=25
2290 IF WY>51 THEN WY=51
2300 IF FX>WX THEN FX=WX;DF=1
2310 IF WY>FY+3 THEN FY=WX-3;DF=1
2320 IF FY>UY THEN FY=UY;DF=1
2330 IF WY>FY+19 THEN FY=WY-19;DF=1
2340 IF FY>32 THEN FY=32
2350 IF DF THEN GOSUB 2090
2360 GOSUB 1990;POKE 2020,32;RETURN
2370 REM
2380 REM +++ UPD CELLA CORRENTE
2390 REM
2400 IF A$="" THEN POKE 2019,32;GOTO 2430
2410 POKE 2019,FNSC(ASC(C$(UX,UY)+ " "));AS=LEFT$(AS,9)
2420 IF PEEK(2019)=22 THEN AS=MID$(STR$(V(WX,WY)),2+(V(WX,WY)<0),9)
2430 PRINT "(HM)(CD)";FOR YS=1 TO (WY-FY+1):PRINT "(CD)";:NEXT:PRINT "(CR)(CR)(CR)";
2440 IF (UX-FX)>0 THEN FOR XS=1 TO UX-FX:PRINT "(CR)(CR)(CR)(CR)(CR)(CR)(CR)(CR)";:NEXT
2450 IF PEEK(2019)<>22 THEN PRINT "(RV)"AS;LEFT$(BL$,9-LEN(AS))"(RD)";
2460 IF PEEK(2019)=22 THEN PRINT "(RV)"LEFT$(BL$,9-LEN(AS))AS"(RD)";
2470 RETURN
2480 REM
2490 REM +++ EDITA ENTRY LINE
2500 REM
2510 XS=1;PRINT "(HM)";
2520 POKE 1023+XS,PEEK(1023+XS) OR 128
2530 GET AS;IF AS="" THEN 2530
2540 POKE 1023+XS,PEEK(1023+XS) AND127
2550 IF AS=CHR$(13) THEN 2640
2560 IF AS="" THEN 2670
2570 IF AS=CHR$(14B) THEN XS=XS-1
2580 IF AS="(DL)" OR AS="(CL)" THEN XS=XS-2
2590 IF XS>79 THEN XS=80;GOTO 2520
2600 IF XS<0 THEN XS=0
2610 IF AS="(CD)" OR AS="(CU)" OR AS=CHR$(34) THEN 2520
2620 POKE216,0;PRINT AS;:XS=XS+1;IF XS>YS THEN YS=XS
2630 GOTO 2520
2640 J=0;POKE 2020,35;AS="";FOR XS=1 TO YS:I=PEEK(XS+1023);AS=AS+CHR$(FNAS(I))
2650 IF I<>32 THEN J=XS
2660 NEXT
2670 AS=LEFT$(AS,J);POKE 2016,FNF(0);RETURN
2680 REM
2690 REM +++ GOTO COORDINATE
2700 REM
2710 PRINT CL$+ER$;GOTO " ";
2720 GET AS;IF AS="" THEN 2720
2730 A=ASC(AS);IF A=13 THEN 2840
2740 IF A<65 OR A>90 THEN 2840
2750 PRINT AS";:XS=A-65
2760 GET AS;IF AS="" THEN 2760
2770 A=ASC(AS);IF A=13 THEN 2840
2780 IF A<48 OR A>57 THEN 2840
2790 PRINT AS;:YS=(A-48)*10
2800 GET AS;IF AS="" THEN 2800
2810 A=ASC(AS);IF A=13 THEN 2840
2820 IF A<48 OR A>57 THEN 2840
2830 PRINT AS;:YS=A-49+YS;WY=XS;UY=YS;GOTO 2220
2840 AS="";PRINT CL$+ER$+"ERRORE IN GOTO";:RETURN
2850 REM ***
2860 REM *** CALCOLO ESPRESSIONI
2870 REM ***
2880 XS=0;I=0;AF%=0;PA%=0;DP%(0)=2;DP(0)=0;ER%=0;FX=0;FX(0)=0
2890 I=I+1
2900 A%=FNX(I);TC%=0;IF A%=32 THEN 2890
2910 IF ER% THEN 4080
2920 IF I>LEN(AS) THEN 4020
2930 FX=0;IF A%<>64 THEN 3150
2940 REM
2950 REM --- CHECK FUNZIONE
2960 REM
2970 FS=""
2980 I=I+1;IF I>LEN(AS) THEN ER%=1;GOTO 4050
2990 IF FNX(I)<>32 THEN FS=FS+CHR$(FNX(I))
3000 IF LEN(FS)<4 AND I<LEN(AS) THEN 2980
3010 IF FS="ABS(" THEN FX=1
3020 IF FS="EXP(" THEN FX=2
```

posiz. cursore: cella nella quale è localizzato il cursore.

Il cursore viene spostato sul tabellone con i normali tasti di controllo. In aggiunta, per semplificare le operazioni, sono abilitati anche questi comandi:

return: porta il cursore in colonna A nella riga successiva a quella in cui si trova al momento

shift + ret: porta il cursore nella prima colonna della finestra e nella riga successiva a quella in cui si trova al momento

home: porta il cursore in A01

shift + home: porta il cursore nella prima casella della finestra

>CRR: funzione GOTO. Porta il cursore nella colonna e riga indicate nel comando (C=colonna, RR=riga).

Tutti i comandi di controllo cursore provocano lo scrolling automatico della finestra del display per consentire l'adattamento alle varie situazioni.

## 2. Caselle e contenuto

Una volta visto come spostare il cursore nell'ambito dell'intero tabellone è ora la volta di provare a modificarne il contenuto. Una casella dello SPREADSHEET può contenere fino ad 80 caratteri. Questi caratteri possono costituire etichette o espressioni.

### Etichette

L'etichetta è una stringa di caratteri con scopo puramente descrittivo. Serve a semplificare la comprensione dei dati del tabellone e non viene manipolata dal programma. Per introdurre un'etichetta nella casella, battere " ". Il programma riconosce il comando con il messaggio 'ENTRY LABEL' nella riga in basso. Poiché il programma visualizza solamente i primi nove caratteri di una etichetta è conveniente limitarne la lunghezza allo stretto necessario, al fine di evitare spreco di memoria. Una cella, comunque, può contenere fino ad 80 caratteri di etichetta che vengono memorizzati anche se non sono costantemente visualizzati sullo schermo.

### Espressioni

La reale potenza di questo programma è nella possibilità di definire in maniera libera correlazioni tra le varie caselle. Queste vengono definite utilizzando nel calcolo del contenuto di una casella, i valori di altre caselle del tabellone. Per fare ciò è sufficiente indicare al programma di voler inserire una espressione battendo il tasto 'V'. Il programma riconosce il comando con il messaggio 'ENTRY VALUE'. A questo punto è possibile introdurre una formula che, elaborata, dia il valore da attribuire alla casella. Questa espressione può essere composta dai quattro operatori matematici, (+, -, \*, /), da funzioni, costanti e riferimenti ad altre celle. Una volta terminata l'introduzione della formula questa viene immediatamente calcolata ed il relativo valore appare sul tabellone. La lun-

```

3030 IF F$="INT(" THEN F%=3
3040 IF F$="LOG(" THEN F%=4
3050 IF F$="LGD(" THEN F%=5
3060 IF F$="SQR(" THEN F%=6
3070 IF F$="SIN(" THEN F%=7
3080 IF F$="COS(" THEN F%=8
3090 IF F$="TAN(" THEN F%=9
3100 IF F$="ATN(" THEN F%=10
3110 IF F$="ASN(" THEN F%=11
3120 IF F$="ACS(" THEN F%=12
3130 IF F%=0 THEN ER%=2:GOTO 4050
3140 GOTO 3450
3150 IF AX<65 OR AX>90 THEN 3260
3160 REM
3170 REM --- CHECK VARIABILE
3180 REM
3190 XS=FNX(I)-65:I=I+1:YS=FNX(I)-48
3200 IFYS<0DRYS>9THENER%=3:GOTO 4050
3210 I=I+1:AX=FNX(I)-48:IFAX<0ORAX>9THENER%=3:GOTO 4050
3220 YS=YS*10+AX:IFYS<10DRYS>52THENER%=4:GOTO 4050
3230 XS=V(XS,YS-1)
3240 GOSUB 3670:IFI=LEN(A$)DRER%THEN 4020
3250 GOTO 2990
3260 IF (AX<48ANDAX<>46)ORAX>57THEN 3440
3270 REM
3280 REM --- CHECK COSTANTE
3290 REM
3300 XS=0:YS=1:I=I-1
3310 I=I+1:AX=FNX(I)-48:IFAX=-2THEN 3350
3320 IF AX<0 OR AX>9 THEN 3400
3330 XS=XS*10+AX:IFI<LEN(A$)THEN 3310
3340 GOTO 4020
3350 I=I+1:AX=FNX(I)-48
3360 IF AX<0 OR AX>9 THEN 3400
3370 YS=YS/10:XS=XS+AX*YS
3380 IF I=LEN(A$) THEN 4020
3390 GOTO 3350
3400 GOSUB 3670:GOTO 2990
3410 REM
3420 REM --- APERTA PARTENTESI
3430 REM
3440 IF AX<>40 THEN 3500
3450 FX(PA%)=FX:PA%=PA%+1:IFPA%>10THENER%=5:GOTO 4050
3460 OP(PA%)=2:OP(PA%)=0:GOTO 2890
3470 REM
3480 REM --- CHIUSA PARTENTESI
3490 REM
3500 IF AX<>41 THEN 3570
3510 PA%=PA%-1:IFPA%<0THENER%=6:GOTO 4050
3520 XS=OP(PA%+1):GOSUB 3670:IFI=LEN(A$)DRER%THEN 4020
3530 GOTO 2890
3540 REM
3550 REM --- OPERATORE ARITMETICO
3560 REM
3570 AF%=0:IF AX=42 THEN AF%=1
3580 IF AX=43 THEN AF%=2
3590 IF AX=45 THEN AF%=3
3600 IF AX=47 THEN AF%=4
3610 IF AX=94 THEN AF%=5
3620 IF AF%=0 THEN ER%=7:GOTO 4050
3630 GOSUB 3670:GOTO 2890
3640 REM
3650 REM --- ESEGUE I CALCOLI
3660 REM
3670 IF F$(PA%) THEN GOSUB 3780
3680 YS=OP(PA%):OP(PA%)=AF%:AF%=0
3690 IF YS=4 AND XS=0 THEN ER%=17:RETURN
3700 IF YS=1 THEN OP(PA%)=OP(PA%)*XS
3710 IF YS=2 THEN OP(PA%)=OP(PA%)+XS
3720 IF YS=3 THEN OP(PA%)=OP(PA%)-XS
3730 IF YS=4 THEN OP(PA%)=OP(PA%)/XS
3740 IF YS=5 THEN OP(PA%)=OP(PA%)^XS
3750 XS=0:RETURN
3760 REM
3770 REM --- ELABORA LE FUNZIONI
3780 REM
3790 ON F$(PA%) GOTO 3810,3820,3840,3850,3870,3890
3800 ON F$(PA%)-6 GOTO 3910,3920,3930,3940,3950,3970
3810 XS=ABS(XS):RETURN
3820 IF XS>99 THEN ER%=11:RETURN
3830 XS=EXP(XS):RETURN
3840 XS=INT(XS):RETURN
3850 IF XS<=0 THEN ER%=12:RETURN
3860 XS=LOG(XS):RETURN
3870 IF XS<=0 THEN ER%=13:RETURN
3880 XS=LOG(XS)/LOG(10):RETURN
3890 IF XS<0 THEN ER%=14:RETURN
3900 XS=SQR(XS):RETURN
3910 XS=SIN(XS*/180):RETURN
3920 XS=COS(XS*/180):RETURN
3930 XS=TAN(XS*/180):RETURN
3940 XS=ATN(XS)*180/:RETURN
3950 IF ABS(XS)>1 THEN ER%=15:RETURN
3960 XS=ATN(XS/SQR(-XS*XS+1))*180/:RETURN
3970 IF ABS(XS)>1 THEN ER%=16:RETURN
3980 XS=-ATN(XS/SQR(-XS*XS+1))*180/+90:RETURN
3990 REM
4000 REM --- CHECK FINALI
4010 REM
4020 GOSUB 3670:XS=0:IFER%THEN 4050
4030 IFPA%THENER%=6:GOTO 4050
4040 RETURN

```

(continua a pag. 136)

ghezza della formula è limitata ad 80 caratteri. Questo valore è sufficiente nella stragrande maggioranza dei casi. Può comunque essere necessario, talvolta, scrivere formule più lunghe. In questo caso sarà necessario scindere la formula in più parti ed utilizzare più caselle per l'elaborazione di un solo dato.

### 3. Comandi

Il programma SPREADSHEET ha un esteso set di comandi che consentono di semplificare l'introduzione, la modifica, il calcolo e la memorizzazione del contenuto del tabellone. Questi comandi ausiliari vengono selezionati con il carattere '/', che provoca l'apparire nella "linea messaggi 2" di un elenco dei comandi disponibili:

COMANDI: B C E G M P S

A questo punto è possibile selezionare il comando desiderato o, premendo RETURN o un tasto non compreso nel set di comandi, ritornare al modo introduzione/modifica dati.

In tutti i casi la pressione del tasto RETURN a livello sottocomandi provoca il ritorno al modo principale senza che il sottocomando venga eseguito.

Alcuni di questi sottocomandi provocano la generazione di un successivo sottocleno (es. /G o /M). Anche in questo caso la pressione di un tasto non compreso nel sottocleno provoca il passaggio al modo introduzione/modifica dati.

Vediamo ora tutti i comandi in dettaglio.

**/B** (blank) — Provoca la cancellazione della cella indirizzata dal cursore. Non vengono modificati i valori di altre celle e non viene ricalcolato il tabellone.

**/C** (cancella) — Il comando Cancella provoca la cancellazione totale di tutti i dati contenuti nel tabellone e dei flag di stato, resettando i valori alla condizione di inizializzazione del programma. Poiché il comando è potenzialmente pericoloso per il contenuto del tabellone, il programma richiede una conferma prima che venga eseguito. Sulla "linea messaggi 2" appare la scritta "CANCELLA: CONFERMA (S/N)". Per confermare battere S. In questo caso il display viene cancellato e il programma reinizializzato. Per evitare la cancellazione battere un tasto diverso da 'S'.

**/E** (edita) — Questo comando permette di editare il contenuto della cella indirizzata dal cursore. Questo viene trasferito nella linea di editing e può essere modificato. Premendo il tasto '←' l'operazione viene abortita ed il contenuto della cella rimane immutato. Se la cella è vuota il comando è inattivo.

**/G** (globale) — Attraverso /G è possibile accedere ad un subset di comandi che permettono di modificare lo stato di funzionamento del programma. Queste modifiche sono permanenti e vengono memorizzate



sentono di gestire la memoria a disposizione del programma:

MEMORIA: A L M

Questo è l'elenco dei sottocomandi:

**/MA** — Abilita la gestione automatica del garbage collection (default). Ogni volta che una cella viene modificata forza un garbage collection. Questo consente di evitare il blocco del programma ogni qualvolta lo spazio di memoria a disposizione delle stringhe viene esaurito, ma produce un ritardo di esecuzione di alcuni comandi direttamente proporzionale al numero di celle avvalorate. È molto utile quando il tabellone non è molto pieno. Se il ritardo diviene sensibilmente avvertibile è opportuno disabilitare questa funzione con il comando **/MM** e forzare manualmente il garbage collection con il comando **/ML**.

**/ML** — Visualizza la memoria ancora disponibile (in byte) e forza un garbage collection. **\*\*\* ATTENZIONE \*\*\*** se il tabellone è molto pieno questo comando può essere eseguito in un tempo abbastanza lungo.

**/MM** — Abilita la gestione manuale del garbage collection. L'abilitazione di questa funzione impedisce al programma di conoscere la memoria ancora disponibile. Nella status line viene quindi visualizzato "?0" al posto del valore percentuale di memoria utilizzabile per i dati. Per conoscere quanto spazio è ancora dedicabile ai dati del tabellone è possibile usare il comando **/ML**.

**/P** (stampa) — Questo comando consente di ottenere l'hardcopy del tabellone contenuto in memoria. Posizionare il cursore sulla prima cella da stampare e battere **'P'**. Il calcolatore richiederà le coordinate dell'ultima cella. Battendo **'return'** il comando viene rifiutato. Vengono stampati anche i riferimenti delle colonne, in reverse.

**/S** (Salva) — Attraverso **/S** è possibile accedere ad un sottoelenco di comandi che consentono di gestire la memorizzazione dei dati del tabellone. Dopo aver battuto il comando, infatti, apparirà la scritta: SALVA: L Q S

Questo è l'elenco dei sottocomandi:

**/SL** — Carica i dati dal supporto di massa selezionato. Non effettua cancellazione del tabellone in memoria, per cui i dati vengono aggiunti a quelli eventualmente esistenti. È così possibile effettuare un Merge tra vari file. Il programma, ovviamente, rimpiazza i dati esistenti in memoria con quelli provenienti dal supporto di massa nel caso vi siano coincidenze con i dati in memoria. Nel caso sia selezionato il disco il programma chiede il nome del file dati da leggere.

**/SQ** — Termina l'esecuzione del programma. Poiché questo comando è potenzialmente pericoloso per i dati contenuti nel tabellone viene chiesta conferma con il messaggio **'QUIT: CONFERMA (S/N)?'**.

Battere **'S'** per terminare il programma, un altro tasto per abortire il comando.

**/SS** — Memorizza i dati sul supporto di massa selezionato. Se è abilitata la registrazione su disco richiede il nome del file da registrare. Prima di dare il comando **/SS** su cassetta è opportuno posizionare il nastro al punto desiderato.

C04+C05+C06+C07+C08+C09+C10				
	A	B	C	D
1	MESE: DTT		%	
2	---			
3				
4	LUCE	40	4.12	
5	AFFITTO	180	18.55	
6	TELEFONO	120	12.37	
7	AUTO	250	25.77	
8	GAS	25	2.57	
9	ACQUA	25	2.57	
10	RATE MOB.	530	54.02	
11	TOTALE	970	99.97	
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				
51				
52				
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				

Ecco invece una copia dello stesso schermo (vedi esempio della pag. a fronte), ottenuta tramite il Simon's Basic. Notate sulla prima riga l'espressione contenuta nella cella C12, consistente nella somma di tutti i contenuti significativi della stessa colonna, va precisato che sullo schermo il cursore e il quadro (contenente le indicazioni delle righe, dalla 1 a 52, e delle colonne, dalla A alla Z) sono in reverse.

Altri comandi non preceduti dalla **'/'**:  
**> (GOTO)** — Posiziona il cursore alla casella indicata. Ad esempio per portare il cursore alla casella J37 battere **>J37**

Valori inesistenti od errati provocano il messaggio **'ERRORE IN GOTO'**.

**!(Calcola)** — Forza il calcolo dell'intero tabellone. Può essere usato quando viene abilitato il calcolo manuale o quando, in calcolo automatico, vi siano dei 'riferimenti in avanti'. Nella definizione del tabellone è possibile, infatti, utilizzare nel calcolo una casella che non è ancora calcolata. Ad esempio introducendo nella casella A01 la formula **'120\*C01/12'** si avrà che il valore di C01 utilizzato nel calcolo dell'espressione sarà diverso da quello visualizzato sul tabellone poiché questo valore viene aggiornato DOPO il calcolo della cella A01. Queste situazioni sono comunemente da evitare poiché sono potenzialmente pericolose e possono portare a risultati imprevedibili. Per evidenziare l'esistenza di 'riferimenti in avanti' è sufficiente forzare il calcolo del tabellone ed osservare se si verificano aggiornamenti di valori. In questo caso è opportuno rianalizzare il problema e modificare il tabellone in maniera da escludere questo tipo di situazione.

#### 4. Espressioni e funzioni

Come abbiamo visto precedentemente la forza principale del programma è nella possibilità di stabilire correlazioni tra celle attraverso l'uso di espressioni. Una espressione è costituita da uno o più dei seguenti elementi:

- costante (numero compreso tra -99999999 e 99999999)
- un riferimento ad una cella espresso in colonna e riga
- un operatore aritmetico (+, \*, -, /, ^)
- una funzione (es. (a SIN)

Nel caso nella espressione venga fatto riferimento ad una cella che contiene una label, questa viene assunta avere valore zero. Il programma effettua il calcolo delle espressioni da sinistra verso destra, senza precedenze. Per stabilirle è necessario utilizzare le parentesi. Ad esempio l'espressione **3 + 12/3\*5** ha come risultato 25 mentre **3 + (12/3\*5)** viene calcolata come 23. Nel caso si verificano errori durante l'elaborazione di un tabellone il programma interrompe il calcolo con la segnalazione della casella in cui si è verificato l'errore e con il tipo di problema incontrato, ad esempio: **'A01: DIVISIONE PER ZERO'**. È sufficiente verificare l'espressione e correggere l'errore. Una cella contenente un errore viene visualizzata sul tabellone con il valore **1E+33**. Da tener presente che, poiché il programma ferma l'esecuzione dei calcoli nel punto in cui viene rilevato un errore, i dati presenti sul tabellone non sono in quel caso attendibili. È necessario rimuovere la causa dell'errore e rielaborare il tabellone. Deve essere posta attenzione ai riferimenti circolari, cioè alle espressioni che richiamano se stesse (direttamente o indirettamente) ad esempio **'A01 + 1'** nella cella A01. Queste situazioni possono compromettere il risultato del programma poiché presentano valori instabili che variano ogni volta che il tabellone viene ricalcolato, anche se non sono state fatte modifiche.

Segue un elenco delle funzioni riconosciute dal programma.

(a SIN(x) seno trigonometrico dell'argomento

(a COS(x) coseno trigonometrico dell'argomento

(a TAN(x) tangente trigonometrica dell'argomento

(a ASN(x) inverso funzione seno

(a ACS(x) inverso funzione coseno

(a ATN(x) inverso funzione tangente

(tutte le funzioni trigonometriche lavorano in gradi sessagesim.)

(a ABS(x) ritorna il valore assoluto dell'argomento

(a EXP(x) ritorna e (2.7182..) elevato all'argomento

(a INT(x) ritorna l'intero più grande minore dell'argomento

(a LGD(x) ritorna il logaritmo in base 10 dell'argomento

(a SQR(x) ritorna la radice quadrata dell'argomento.

Questo programma è disponibile su cassetta presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 120.