



Da questo numero della rubrica di software per calcolatrici con Sistema Operativo Algebrico, inizieremo a pubblicare soprattutto programmi inviati dai lettori: confessiamo che la scelta non è facile e cercheremo di essere imparziali nello scegliere programmi il più possibile interessanti, ben realizzati e ben presentati. L'esperienza insegna che è praticamente impossibile ottenere un interesse del 100% da parte dei lettori, speriamo comunque di accontentarne ogni volta il maggior numero possibile.

## Macchina per scrivere

di Paolo Monterotti - Roma

Come ci si potrà accorgere subito osservando il listing, il programma è molto semplice, ma nella sua semplicità assolve completamente ed elegantemente una funzione che forse è l'unico neo nel complesso e sofisticato insieme di operazioni possibili con le TI 58 e 59: la stampa di caratteri alfanumerici.

Risulta infatti alquanto macchinoso il metodo da seguire per stampare qualcosa con la printer PC 100C: nulla di trascendentale, intendiamoci, basta un minimo di pratica per imparare il meccanismo.

In particolare ad ogni carattere alfanumerico o speciale viene associato un codice di due cifre, secondo uno schema a matrice rappresentato in figura 1. Per stampare una riga di 20 caratteri bisogna riempire 4 buffer (01, 02, 03, 04) ognuno composto da 5 caratteri e per fare ciò bisogna impostare i codici dei 5 caratteri, premere il tasto 2nd Op e successivamente il numero del buffer. Per stampare poi il contenuto dell'intera riga di stampa si deve premere Op 05, mentre con Op 00 otterremo la cancellazione dei buffer.

Il programma "Macchina per scrivere" del lettore Paolo Monterotti di Roma consente di costruire con semplicità la riga di stampa, senza dover sempre consultare la tabella di fig. 1: diciamo subito che un programma simile si trova nel modulo Solid State Software M.U. (Math/Utilities) ma presenta caratteristiche completamente differenti e soprattutto un meccanismo un po' più pesante.

Il programma proposto permette di associare praticamente ad ogni tasto un carattere alfanumerico, proprio come in una macchina per scrivere.

È interessante notare che non è necessario premere il tasto SBR prima di quello che corrisponde alla lettera desiderata: ciò è dovuto al fatto che all'ultimo passo della memoria di programma vi è un'istruzione

### CODICI ALFANUMERICI

	01234567	89
0	0123456	78
1	789ABCDE	78
2	-FGHIJKL	MN
3	MNOPQRST	MN
4	-UVWXYZ+	XX
5	X*Y/e(O)	XX
6	↑%:/=^*%Z	89
7	89÷!L.ATZ	89
8	0123456	78
9	789ABCDE	78

FIG. 1

SBR che, quando incontrata durante l'elaborazione, provoca l'arresto della calcolatrice con conseguente lampeggiamento del display.

In questo stato la calcolatrice riconosce il primo tasto premuto come il seguito dell'istruzione interrotta e riprende automaticamente l'elaborazione, mantenendo però il lampeggio del visualizzatore. Basterà a questo punto prevedere, in ogni subroutine chiamata, la cancellazione di questo lampeggiamento.

Il programma ha il suo nucleo centrale nella sequenza da 000 a 063, che ha lo scopo di costruire le righe di stampa riempiendo opportunamente i buffer, mentre a seguire devono essere inserite le etichette con i codici di stampa desiderati: Lbl XXX NN RST, dove XXX è il nome della label ed NN è il codice alfanumerico.

A questo punto "confessiamo" di aver apportato una piccolissima correzione al programma di Monterotti, che consente di risparmiare una buona dose di passi di programma: vediamo ancor più da vicino il funzionamento per capire il significato della correzione proposta.

MACCHINA	027	02	02	060	00	00	093	81	RST		
PER	028	69	DP	061	61	GT0	094	76	LBL		
SCRIVERE	029	22	22	062	02	02	095	15	E		
	030	25	CLR	063	37	37	096	01	1		
	031	42	STD	064	76	LBL	097	07	7		
	032	01	01	065	99	PRT	098	81	RST		
000	95	=	033	08	8	066	43	RCL	099	76	LBL
001	24	CE	034	42	STD	067	01	01	100	22	INV
002	65	%	035	00	00	068	84	DP+	101	02	2
003	01	1	036	05	5	069	02	02	102	01	1
004	00	0	037	32	XIT	070	69	DP	103	81	RST
005	45	Y%	038	43	RCL	071	05	05	104	76	LBL
006	43	RCL	039	02	02	072	61	GT0	105	23	LNW
007	00	00	040	22	INV	073	10	E*	106	02	2
008	95	=	041	67	EQ	074	76	LBL	107	02	2
009	52	EE	042	02	02	075	11	A	108	81	RST
010	22	INV	043	37	37	076	01	1	109	76	LBL
011	52	EE	044	69	DP	077	00	3	110	24	CE
012	44	SUM	045	05	05	078	81	RST	111	02	2
013	01	01	046	69	DP	079	76	LBL	112	03	3
014	69	DP	047	00	00	080	12	B	113	81	RST
015	30	30	048	76	LBL	081	01	1	114	76	LBL
016	69	DP	049	10	E*	082	04	4	115	25	CLP
017	30	30	050	25	CLR	083	81	RST	116	02	2
018	43	RCL	051	42	STD	084	76	LBL	117	04	4
019	00	00	052	01	01	085	13	C	118	81	RST
020	29	DP	053	01	1	086	01	1	119	76	LBL
021	77	GE	054	42	STD	087	05	5	120	32	XIT
022	02	02	055	02	02	088	81	RST	121	00	0
023	37	37	056	08	8	089	76	LBL	122	00	0
024	43	RCL	057	42	STD	090	14	D	123	00	0
025	01	01	058	00	00	091	01	1	237	43	RCL
026	84	DP+	059	69	DP	092	06	6	239	01	01
									239	71	SBR



Abbiamo visto prima che premendo un tasto di funzione la calcolatrice completa l'istruzione di chiamata a subroutine, mantenendo la condizione di display lampeggiante: ora per eliminare questo stato Monterotti proponeva di inserire, nella sequenza vista prima, un CLR tra XXX ed N, dal momento che CLR elimina la condizione di display lampeggiante cancellando il contenuto. Ora però questa istruzione ripetuta in ogni etichetta peggiora l'occupazione di memoria del programma stesso: fatti i conti, con 26 lettere dell'alfabeto più le 10 cifre più ancora tutti i caratteri speciali si arriva, a volerli usare tutti, a 64 caratteri e perciò a 64 istruzioni CLR.

Osservando meglio ogni etichetta si vede che ogni "frammento" finisce con un RST, il che fa sì che il programma, ogni volta, ritorna al passo 000: è proprio in questo punto che possiamo inserire delle istruzioni che fermano il lampeggio senza però cancellare il contenuto del display, che a questo livello è il codice alfanumerico desiderato. Una sequenza sicura è per l'appunto " = CE" che è stata aggiunta ai passi 000 e 001. Infine un'altra aggiunta è l'etichetta Prt che consente di stampare la riga fin lì costruita.

Nel listing sono riportate come esempio delle etichette corrispondenti alle prime lettere dell'alfabeto. Bisognerà inoltre prevedere un'etichetta (e quindi un tasto) per lo spazio bianco.

Insomma lasciamo al lettore la più ampia libertà di porre le lettere nell'ordine desiderato, nonché di aggiungere le routine più svariate, ad esempio quella che permette di iniziare la stampa da colonna N senza dover premere per N-1 volte il tasto corrispondente allo spazio bianco.

A questo proposito ricordiamo che non si possono usare i tasti 2nd LRN SST BST come etichette, mentre si possono usare anche le seconde funzioni premendo ovviamente il tasto 2nd prima della funzione stessa. Altra possibilità, per l'etichetta Prt (che fa stampare la riga di stampa) è di premere direttamente il tasto PRINT presente sulla stampante invece di 2nd Prt sulla tastiera. Concludiamo segnalando che il programma utilizza soltanto i registri 00, 01, 02 (lasciando così la più ampia libertà per il numero di passi utilizzati), non usa alcun flag mentre per la ripartizione della memoria bisogna regolarsi vedendo qual è il numero di passi usati, in funzione dei "tasti" della macchina per scrivere.

È fondamentale però, qualunque sia la ripartizione usata, porre negli ultimi tre passi di programma la sequenza RCL 01 SBR ed inoltre bisogna inserire l'indirizzo dell'istruzione RCL (nel nostro caso 237) nei passi 22-23, 42-43, 62-63 del programma altrimenti non si avrà il funzionamento desiderato.

### Come si usa il programma

Il funzionamento è molto semplice: introdotto il programma (attenzione alla ripartizione ed alle considerazioni preceden-

ti) e tutte le etichette che si riferiscono ai "tasti" desiderati, basta premere E', che serve in qualsiasi momento a cancellare la riga di stampa eventualmente impostata e ad inizializzare il tutto.

Dopo nemmeno un secondo il display lampeggerà e successivamente, ogni volta che vedremo il display lampeggiare, potremo premere i tasti a seconda di ciò che vogliamo scrivere.

Automaticamente, non appena i 4 buf-

fer saranno riempiti, si avrà la stampa della riga, con successiva cancellazione dei buffer stessi: a chi non andasse bene questa cancellazione consigliamo di prevedere una memorizzazione in registri successivi del contenuto dei buffer mano a mano che vengano riempiti. Consigliamo infine di costruirsi una mascherina (di carta o di plastica) da porre sopra la tastiera, per avere un'indicazione della corrispondenza delle lettere con i tasti della calcolatrice.

MC

## L'angolo delle T.I.

*Crediamo di far cosa gradita ai lettori che non conoscono il funzionamento delle calcolatrici TI 58 e 59, spiegando il significato di un'istruzione speciale, l'Op.*

*È questa una funzione particolare che permette alla calcolatrice di compiere un certo numero di operazioni eterogenee, premendo il tasto Op ed un codice di due cifre NN; per i possessori di calcolatrici Texas aggiungiamo che vi sono ulteriori informazioni riguardanti questa istruzione, dettate dall'esperienza.*

*Per comodità indicheremo come capoverso il valore del codice NN:*

00 - Cancella il contenuto dei 4 buffer di stampa

01, 02, 03, 04 - Inserisce nel buffer indicato i codici alfanumerici presenti nel visualizzatore

05 - Stampa il contenuto dei buffer in una riga di stampa

06 - Stampa gli ultimi quattro caratteri del buffer 04 sulla destra del contenuto del display

07 - Stampa un asterisco nella colonna da 0 a 19 a seconda del valore impostato sul display

08 - Stampa la lista delle etichette contenute nel programma

09 - Consente di trasferire un programma della biblioteca S.S.S. nella memoria della calcolatrice previa predisposizione della opportuna ripartizione della memoria a seconda della lunghezza del programma da memorizzare

10 - Funzione sign (x): se il display conteneva un numero positivo si avrà 1, se negativo si avrà -1, se nullo si avrà 0

11, 12, 13, 14, 15 - Operazioni statistiche che permettono, una volta impostati i dati da analizzare, di calcolare la varianza (11), la deviazione standard (INV Op 11  $\sqrt{x}$ ) e la retta passante per i punti dati secondo la regressione lineare; di questa retta vengono forniti: la pendenza e l'intersezione con l'asse delle ordinate (12), il coefficiente di correlazione (13) e i valori stimati dall'ordinata per valori impostati della ascissa (14) e viceversa (15)

16 - Visualizza la ripartizione di memoria con il numero XXX.YY dove XXX rappresenta l'ultimo passo di programma disponibile e YY è il valore del più alto registro dati

17 - Predisporre la ripartizione della memoria considerando il valore presente sul display come numero di blocchi di memoria desiderati; ognuno di questi blocchi è formato da 10 registri

18 - Attiva il flag 7 se, al momento in cui è incontrata l'istruzione, la calcolatrice non è in condizione d'errore. INV Op 18 invece resetta il flag 7 se non vi è errore

19 - Setta il flag 7 se la calcolatrice si trova in stato di errore quando incontra l'istruzione. INV Op 19, analogamente a prima, resetta il flag 7 se si è in condizione d'errore

20-29 - Funzione utilissima in quanto consente di incrementare di un'unità il contenuto del registro 0-9: ad es. Op 25 incrementa il contenuto del registro 05

30-39 - Analoga alla precedente solo che decrementa i registri.

*Richiamiamo l'attenzione in particolare su queste due ultime funzioni, OP 2N e 3N in quanto molto spesso, inespugnabilmente, non vengono utilizzate (come se non fossero conosciute): capita infatti di vedere sequenze del tipo 1 SUM 00 o ancora 1 INV SUM 00, formati rispettivamente da 3 e 4 byte, quando invece si possono usare con tutta tranquillità le Op 20 e Op 30 che entrambe sono formate da 2 byte.*

*Ma quel che è peggio è che i programmi della biblioteca Master sono costellati di queste inutili sequenze: tra l'altro impostando 1 SUM 00, quell'1" cancella il precedente contenuto del display, cosa che ovviamente non succede con le Op.*

*Ad esempio è del tutto inspiegabile la sequenza presente nei passi 097-104 del Pgm 02 della "Master Library":*

$x \leftarrow 1 \text{ SUM } 01 \quad x \leftarrow 1 \text{ GT } 093$

*A parte i tre passi sprecati, costringe l'uso dell'istruzione "x←1" per ben due volte per non perdere il contenuto del display! Forse che l'istruzione Op 2N o 3N, per qualche strano motivo "hardware" non funziona se incontrata in un programma della biblioteca?*

*Tornando alle Op 00-08, si ha che queste funzioni "sentono" se la calcolatrice è connessa alla stampante (non importa se accesa o spenta!). In ogni caso le Op 00-05 e 07 considerano solo la parte intera del numero visualizzato sul display e modificano conseguentemente il display stesso, come se si fosse incontrata l'istruzione Int.*

*Se la calcolatrice è da sola le Op 01-04 non riempiono i buffer di stampa corrispondenti: questa affermazione, anche se può sembrare ovvia, assumerà un preciso significato in base a quanto esporremo nel prossimo numero, in cui vedremo ulteriori caratteristiche delle calcolatrici Texas Instruments.*