

## Integrali doppi

di Daniele Tincani - Retignano (LU)

Questo programma calcola l'integrale di una funzione continua di due variabili  $F(x,y)$ , esteso ad un dominio  $D$  "normale" rispetto ad almeno uno degli assi cartesiani.

Ricordiamo che un dominio si dice "normale" rispetto, ad esempio, all'asse  $x$  se è costituito di punti  $P(x,y)$  tali che per  $x_0 \leq x \leq x_n$  esiste un punto  $y$  soddisfacente alla condizione  $f_{inf}(x) \leq y \leq f_{sup}(x)$ . Analogamente si ha per l'asse  $y$ . Facciamo d'ora in poi riferimento al primo caso, rappresentato in figura 1.

Consideriamo perciò un dominio  $D$  normale all'asse  $x$  ed una funzione  $F(x,y)$  continua in  $D$ . Vogliamo calcolare

$$I = \iint_D F(x,y) dx dy$$

Valendoci della formula "di riduzione" possiamo scrivere:

$$I = \int_{x_0}^{x_n} \left[ \int_{f_{inf}(x)}^{f_{sup}(x)} F(x,y) dy \right] dx$$

Allora scelti  $(n_x + 1)$  punti dell'intervallo  $x_0, x_n$ , distanti uno dall'altro di una grandezza  $\Delta x = (x_n - x_0) / n_x$ , (chiamiamo questi punti con  $x_i$ , per  $0 \leq i \leq n_x$ ), potremo calcolare, valendoci del Pgm 09 della "biblioteca di base",  $n_x + 1$  integrali del tipo

$$I_i = I(x_i) = \int_{f_{inf}(x)}^{f_{sup}(x)} F(x_i, y) dy$$

### INVIATECI 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.

Potremo poi calcolare  $I$  tramite il Pgm 10 della biblioteca di base come

$$\int_{x_0}^{x_n} I(x) dx$$

dove

$$I(x) = \int_{f_{inf}(x)}^{f_{sup}(x)} F(x,y) dy$$

è una funzione con valori  $I_i$  a noi nota in  $n_x + 1$  punti.

### Il programma

Prima di iniziare l'analisi del programma, consigliamo i lettori di tenere sottomano il manuale d'uso della "biblioteca di base" per avere maggiori dettagli su quanto scriveremo.

Il programma è fondamentalmente diviso in due parti distinte: nella prima (Lbl A, B,

funzione, rimandiamo ai numeri precedenti di MCmicrocomputer, in particolare modo nell'ambito dell'"Angolo delle TI": ci limitiamo a ricordare che un'istruzione del tipo "HIR mn" si introduce con la sequenza:

STO 82 STO mn BST BST Del BST BST Del

seguita da SST SST per continuare con l'impostazione delle istruzioni successive.

Un'altra istruzione "speciale" è quella che compare al passo 097: con il codice 31 si ottiene il passaggio al modo di apprendimento (LRN mode) direttamente da programma.

Ciò fa sì che, all'atto dell'introduzione dei parametri iniziali del calcolo dell'integrale, la calcolatrice passa automaticamente in LRN per permetterci l'impostazione delle tre funzioni  $f_{inf}$ ,  $f_{sup}$  ed  $F$ .

Analogamente a quanto visto per l'HIR, il codice 31 si introduce con la sequenza STO 31 BST BST Del SST.

Il programma è previsto per tutti e tre i modelli di TI (58, 58C e 59), dato che il programma è di appena 98 passi, più quelli delle label B' ed A'.

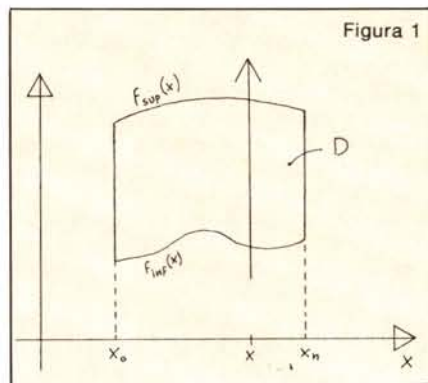


Figura 1

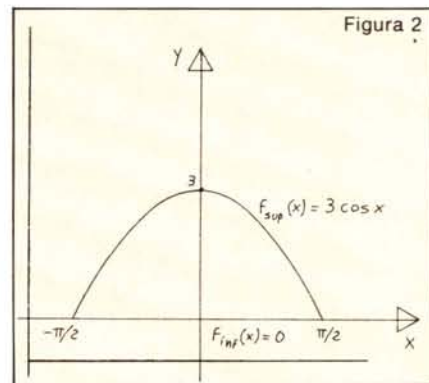


Figura 2

C, D) avviene il caricamento di tutte le informazioni, comprese le istruzioni per il calcolo di  $f_{inf}(x_i)$  e  $f_{sup}(x_i)$  (Lbl B') e per il calcolo di  $F(x_i, y)$  (Lbl A').

Nella seconda (Lbl E), viene calcolato l'integrale col metodo visto precedentemente.

L'impiego massiccio dei registri "interni", introdotti dal codice 82 (HIR) è risultato necessario per reperire registri dati senza ricorrere ad acrobatici indirizzamenti indiretti.

Si è evitato però il loro impiego nella specificazione delle tre funzioni  $f_{inf}$ ,  $f_{sup}$  ed  $F$  (Lbl B' ed A'), per non compromettere troppo la maneggevolezza del programma stesso.

Per i dettagli riguardanti l'uso di questa

Per quanto riguarda i registri, in particolare, i Pgm 09 e Pgm 10 usano R00 ... R05 per i parametri di calcolo, nonché i registri di R06 fino al massimo consentito dalla ripartizione per la memorizzazione dei valori  $I_i$ , usati per l'appunto dal Pgm 10.

Per ciò che riguarda tutte le limitazioni sui valori dei parametri iniziali, si rimanda alle norme d'uso dei due programmi della biblioteca: in particolare ricordiamo che i valori  $n_x$  ed  $n_y$  devono essere pari.

Infine non è previsto l'uso della stampante, in quanto il risultato è ottenibile direttamente sul display. In caso di connessione con la stampante si avrà, verso la fine del calcolo, la stampa del valore  $n_y$  e successivamente del valore  $I$ : tale stampa è

automatica in quanto inglobata all'interno del Pgm 10.

Un'ultima annotazione per i registri HIR: vengono usati i registri H3, H4, H5, H6 ed H7, più che altro per l'"appoggio" di dati che servono successivamente nel corso dell'elaborazione.

**Uso**

Facciamo riferimento alla figura 2, e calcoliamo

$$I = \int_{-\pi/2}^{\pi/2} dx \int_0^{3\cos x} y^2 \sin^2 x dy$$

calcolato con  $n_x = n_y = 10$ .

A questo scopo impostiamo il valore  $x_0$  ( $= -\pi/2$ ) e premiamo A; impostiamo  $x_n$  ( $= \pi/2$ ) e premiamo B; impostiamo  $n_x$  ( $= 10$ ) e premiamo C ed infine impostiamo il valore di  $n_y$  ( $= 10$ ) e premiamo D. A questo punto il visualizzatore mostrerà che la calcolatrice è entrata in modo LRN mostrandoci il passo 098: introduciamo perciò le tre funzioni con le seguenti regole.

Per le  $f_{inf}$  e  $f_{sup}$  (Lbl B') entrambe "attengono" il valore della  $x$  dal registro 00 ed il loro valore calcolato deve essere posto rispettivamente in STO 01 e STO 02. Alla fine di tali calcoli bisogna porre GTO C'.

Di seguito si imposta la  $F(x,y)$  (Lbl A'): innanzitutto bisogna memorizzare in STO 02 il valore della  $Y$  proveniente dal pro-

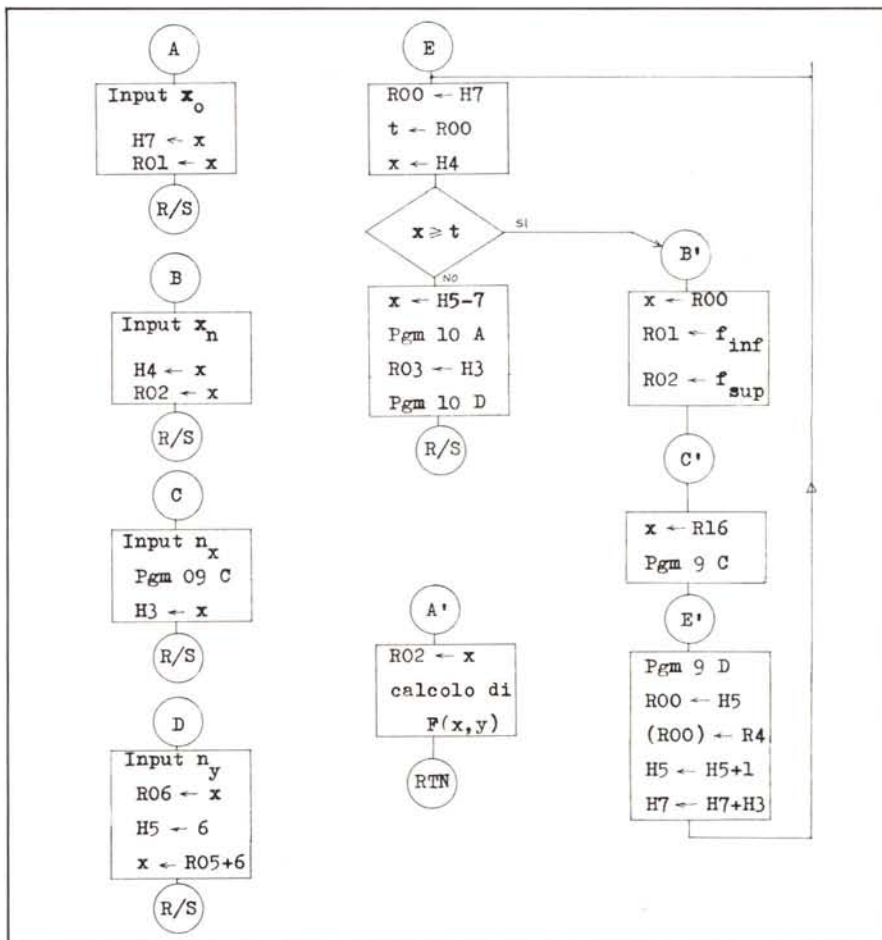
098	76	LBL	118	53	(
099	17	B'	119	24	CE
100	70	RAD	120	65	*
101	00	0	121	43	RCL
102	42	STO	122	00	00
103	01	01	123	38	SIN
104	43	RCL	124	54	)
105	00	00	125	33	X <sup>2</sup>
106	39	CDS	126	92	RTN
107	65	*	127	00	0
108	03	3	128	00	0
109	95	=	129	00	0
110	42	STO			
111	02	02			
112	61	GTO			
113	18	C'			
114	76	LBL			
115	16	A'			10.
116	42	STO			
117	02	02	2.391339189		

Segmento da aggiungere al programma per il calcolo dell'integrale nell'esempio riportato nel testo.

gramma principale all'atto della chiamata alla subroutine A'.

Quindi si imposta la funzione delle due variabili (la  $x$  è sempre in R00) stando attenti a non usare durante i calcoli i tasti = e CLR, ma viceversa effettuando i calcoli all'interno di coppie di parentesi (così

Integrali doppi											
000	76	LBL	028	05	05	056	42	STO	084	35	35
001	11	A	029	85	+	057	03	03	085	82	HIR
002	82	HIR	030	43	RCL	058	36	PGM	086	17	17
003	07	07	031	05	05	059	10	10	087	85	+
004	42	STO	032	95	=	060	14	D	088	82	HIR
005	01	01	033	61	GTO	061	91	R/S	089	13	13
006	91	R/S	034	19	D'	062	76	LBL	090	95	=
007	76	LBL	035	76	LBL	063	18	C'	091	82	HIR
008	12	B	036	15	E	064	82	HIR	092	07	07
009	82	HIR	037	82	HIR	065	16	16	093	61	GTO
010	04	04	038	17	17	066	36	PGM	094	15	E
011	42	STO	039	42	STO	067	09	09	095	76	LBL
012	02	02	040	00	00	068	13	C	096	19	D'
013	91	R/S	041	32	X/T	069	76	LBL	097	31	LRN
014	76	LBL	042	82	HIR	070	10	E'	098	00	0
015	13	C	043	14	14	071	36	PGM	099	00	0
016	36	PGM	044	77	GE	072	09	09	100	00	0
017	09	09	045	17	B'	073	14	D			
018	13	C	046	82	HIR	074	82	HIR			
019	82	HIR	047	15	15	075	15	15	001	11	A
020	03	03	048	75	-	076	42	STO	008	12	B
021	91	R/S	049	07	7	077	00	00	015	13	C
022	76	LBL	050	95	=	078	43	RCL	023	14	D
023	14	D	051	36	PGM	079	04	04	036	15	E
024	82	HIR	052	10	10	080	72	ST*	063	18	C'
025	06	06	053	11	A	081	00	00	070	10	E'
026	06	6	054	82	HIR	082	01	1	096	19	D'



Flow-chart del programma "Integrali doppi"

# L'ANGOLO DELLE TI L'ANGOLO DELLE TI L'AN

Riprendiamo in questo numero il discorso, introdotto nel n° 10 di MCmicrocomputer, riguardante il sistema operativo delle calcolatrici Texas Instruments.

Prima però di affrontare nuovi argomenti, desideriamo fare una precisazione, esplicitamente richiesta dall'interessato: nel già citato n° 10 abbiamo parlato alquanto in dettaglio di alcune istruzioni della calcolatrice, che per il loro funzionamento necessitano di un vero e proprio programma, ormai facilmente accessibile, seppur con metodi "strani".

Ebbene la maggior parte delle informazioni riportate è stata ricavata da una lettera del nostro lettore Alessandro Triglia.

Perciò, dopo questa piccola e doverosa precisazione (... date a Cesare quel che è di Cesare ...), torniamo alla questione: questa volta prendiamo le notizie da una monumentale lettera del simpatico lettore Stefano Laporta di Bologna, il quale continua impertentito nella sua caccia ai segreti della sua malcapitata TI-58C. Abbiamo pensato di dividere tale lettera in più numeri, anche per non "mettere troppa carne al fuoco" tutta in una sola volta.

Nulla toglie che ai lettori è vivamente richiesto di collaborare a questa mini rubrica, che tra l'altro, in base alle lettere pervenute, sta riscuotendo gran successo presso i "SOAisti".

Lasciamo dunque la parola al nostro "vecchio" lettore.

"Prima di tutto una correzione: se la mia TI-58C non è impazzita, la sequenza "codice 21 - funzione trigonometrica" non ha altro effetto che ... calcolare la funzione trigonometrica: probabilmente si tratta di diversità circuitale fra 58C e 59, che rimedia a tale "buco" programmatico.

Riguardo all'analisi del programma contenuto nella ROM devo fare alcune precisazioni nonché aggiungere qualcosa.

1) Del contenuto visibile della ROM (576 passi) ne vengono utilizzati come programma 380 e non 370 come riportato sul n° 10. Gli altri passi (384-575) contengono alcune costanti fondamentali usate dalla calcolatrice, e che è facile riconoscere dopo aver (pazientemente) copiato in RAM tali passi, cambiato partizione ed esaminato i registri a cui corrispondono i passi sopracitati.

Si scopre così che la TI "conosce" i seguenti numeri:

passi	384-391	ln 10
	392-399	ln 2
	448-455	$\pi/4$
	488-495	$\pi/2$
	496-503	$\pi$
	504-511	$180/\pi$

Tutte queste costanti però sono moltiplicate per varie potenze di 10; gli altri passi contengono probabilmente indirizzi o costanti non riconoscibili.

2) Ho notato che il programma corrispondente alla funzione Op 12 comincia al passo 000 e non al passo 002; difatti viene eseguito un HIR 08 "inutile", come è semplice vedere facendo 999 Op 12 HIR 18.

che restituisce appunto il valore 999, "catturato" dall'HIR 08.

3) Strani caratteri: facendo fare alcune pazzie alla 58, si vedono talvolta comparire strani simboli tipo virgolette, apostrofo, gradi, ecc. ... Non sono assurdità, bensì veri e propri caratteri che hanno un loro preciso valore numerico.

Per vederlo effettuiamo alcune operazioni: riempiamo la memoria di programma con codici uguali al rispettivo passo di programma (cioè 000 00, 001 01, ... sino a 040 40); portiamoci al passo 036, ritorniamo in modo esecuzione e impostiamo questa importante sequenza:

26 STO 00 Pgm 01 A Pgm 12 A LRN

senza preoccupazioni se quasi subito vedremo lampeggiare il display. Vedremo nel visualizzatore -83707353"3!!! Cosa significa? La sequenza riportata dà nel display una immagine della memoria di programma nei pressi del passo 036, ovvero fornisce il contenuto byte per byte del registro di memoria a cui apparterebbe il passo 36 se la ripartizione fosse diversa: in questo caso il registro 55, dai passi 032 a 039, che contiene byte per byte 3938373635343332.

Dato che ogni registro contiene 16 cifre e nel display viceversa ci sono solo 11 cifre più il posto del segno, mancheranno 2 cifre a sinistra (il byte 39) e 2 a destra (il byte 32); inoltre il "3" del byte 38 viene visualizzato con il segno "-" per ovvi motivi fisici ed il passo da cui eravamo partiti, il passo 036, è rappresentato con 07 (non ho ancora capito perché ...).

La cosa più importante è che ci sono due apostrofi a destra; cambiando passo di partenza si ottengono altri caratteri; partendo da 034 si ha -83736350"3, mentre da 044 si ha 7000°. Dopo una profonda riflessione ho capito che la TI rappresenta con questi caratteri il passo dal quale si è partiti: più precisamente, chiamando con P il passo di partenza ed essendo N i caratteri "strani", i primi N-1 partendo da sinistra rappresentano il valore  $\text{Int}(P/8)$  e l'N-esimo, che occupa sempre il penultimo display da destra, è il resto di tale divisione e cioè vale  $(P \bmod 8)$ .

Ad esempio se  $P=044$ , si ha  $\text{Int}(44/8) = 5$  corrispondente al simbolo "gradi" e  $44 \bmod 8 = 4$  corrispondente all'apostrofo: ecco perciò svelato (in parte) il mistero.

Ho trovato dunque questa corrispondenza

virgolette	2
blank	3
apostrofo	4
gradi	5
meno	6

mentre i caratteri 0, 1, 7, 8 e 9 rimangono inalterati.

Faccio un altro esempio: voglio vedere tre apostrofi sul display: dato che tale simbolo corrisponde a 4, tre apostrofi vogliono dire "444", che tradotti valgono  $44 \cdot 8 + 4 = 356$ .

Mi porto dunque al passo 356 (non importa che programma c'è in memoria) e nel modo di esecuzione premo 26 STO 00 Pgm 1 A Pgm 12 A LRN: appariranno i tre apostrofi (EUREKA).

è prescritto nel manuale d'uso dei Pgm usati ...). Il tutto deve essere concluso da INV SBR.

Nel listato riportiamo per l'appunto due subroutine separate, in quanto cambieranno al variare dell'integrale che dobbiamo calcolare.

Introdotte perciò tali subroutine, si preme LRN ed il display mostrerà il numero di registri necessari alla calcolatrice per

l'elaborazione.

Premiamo ora E per far partire il calcolo vero e proprio ed aspettiamo ... Il tempo di elaborazione aumenterà all'aumentare dei valori  $n_x$  ed  $n_y$ , mentre viceversa aumenterà la precisione del calcolo ottenuto.

Nel nostro caso in particolare otteniamo un valore di 2.39133 rispetto al valore di 2.4 ottenibile analiticamente.

Chiudiamo questa descrizione con un

appuntamento al nostro lettore e che giriamo agli altri lettori.

Ricordiamo che se nel corso dell'elaborazione si trova ad esempio "A" la calcolatrice salterà al sottoprogramma etichettato con Lbl A, e perciò (come per una qualsiasi altra subroutine) risulterà memorizzato l'"indirizzo di ritorno della subroutine", al quale ritornerà il programma, nel caso incontri INV SBR. Capita spesso però che

# SOLO DELLE TI L'ANGOLO DELLE TI L'ANGOLO

Consiglio di provare anche i passi 178, 267, 445, 534.

Alla luce di queste scoperte si possono fare tre considerazioni: la prima è che questi 6 caratteri strani potrebbero corrispondere ai valori 10-15 e perciò alle cifre esadecimali A-F; probabilmente ne riparleremo in un prossimo numero.

La seconda considerazione è che si può vedere dove è fermo il contatore di programma, anche se il passo corrispondente non è raggiungibile con LRN perché la ripartizione non lo permette.

Ad esempio se partiamo (con 0 Op 17) dal passo 356 e subito dopo portiamo la partizione a "79.49" (con 5 Op 17), premendo anche più volte il tasto LRN non è possibile passare in modo di apprendimento: usando viceversa la sequenza sopra descritta otterremo comunque i tre apici, corrispondenti perciò al passo 356.

La terza considerazione è che questi caratteri vengono usati dalla calcolatrice! Il segno meno viene usato normalmente, il blank spegne il display dove necessita, come fra mantissa ed esponente oppure tra contatore di programma e contenuto del passo. Per i caratteri rimanenti (° ' ") ho una mia teoria: suppongo che questi servissero per visualizzare nell'ultimo display a sinistra la notazione angolare, con la corrispondenza:

- ° Deg
- ' Rad
- " Grad

in maniera analoga a quanto succede ad esempio nella TI-30.

Successivamente alcuni probabili problemi hardware hanno fatto eliminare questa funzione, pur rimanendo tali caratteri disponibili all'interno della TI, forse in maniera analoga a quello che è successo per le HIR.

In alcuni casi però il meccanismo di codifica in pseudo-ottale sembra non funzionare a dovere, apparentemente segnalando passi di programma inconsistenti: comunque anche di questo parleremo in un prossimo numero.

4) Vediamo ora altre sequenze analoghe a 26 STO 00 Pgm 1 A. Sono riuscito, come mi ero ripromesso, ad ottenere dalla mia (... povera ...) 58 comportamenti veramente originali: ho semplicemente acceso e spento la TI almeno 10000 volte con un piccolo circuito elettrico (sconsiglio però vivamente i lettori di imitarmi ...).

Ad esempio ho ottenuto una ripartizione 0., una 2739.39, una 799999999 e così via, ma soprattutto ho scoperto che la RAM della TI utilizzabile per i programmi è formata da 512 byte!!! Ho approfittato di questo per copiare con Op 09 (che, strano a dirsi, funzionava!) i primi 512 passi dei programmi Pgm 02 e Pgm 19' che risultano alquanto misteriosi per un possessore di TI-58.

Ho scoperto così che fanno largo uso di indirizzamento diretto e perciò mi sono messo alla ricerca di passi di programma che potessero far eseguire operazioni "proibite", all'interno del modulo di biblioteca: mi spiego meglio.

Se nel modulo c'è una sequenza "GTO 136 RCL 04" (i cui

codici sono 61 01 36 43 04), posso sfruttare i passi contenenti i codici 36 e 43 per ingannare la calcolatrice e farle eseguire un "Pgm 43": tale sequenza è "proibita" in quanto in realtà nel modulo vi sono 25 programmi. Però mentre impostando da tastiera "Pgm 43" la calcolatrice segnala subito l'errore, rifiutando di accettare altri comandi se non "sensati", viceversa se tale sequenza è incontrata all'interno del modulo succedono fatti strani.

Analizzando il Pgm 02 (almeno per i primi 512 passi) ho scoperto che al passo 056 c'è la sequenza 69 75 corrispondente ad Op 75: ora SBR 056 genera semplicemente un lampeggio, anche se ciò non era ovvio.

Analizzando invece il Pgm 19 si trova che dal passo 042 c'è la sequenza

Ifflg 3 136 Ifflg 4 136 (CE)

i cui codici sono rispettivamente

87 03 01 36 87 04 01 36 53 24

e perciò SBR 045 genera un "Pgm 87" (codici 36 87) e SBR 049 genera un "Pgm 53" (codici 36 53), entrambi "proibiti".

Ora si ha che (ed anche questo non era predicibile) le sequenze "Pgm 19 SBR 045" e "Pgm 19 SBR 049" sono del tutto equivalenti alla "26 STO 00 Pgm 1 A", con il vantaggio di non sprecare un registro di memoria, utile nel caso si operi con la ripartizione 0 Op 17.

Al passo 172 invece c'è la coppia di codici 84 53 corrispondenti ad "Op Ind 53"; usando la ripartizione 6 Op 17 e mettendo vari valori nel registro 53 si può vedere, con "Pgm 19 SBR 172", che tutte le funzioni Op possono essere eseguite in un programma di biblioteca, come ad esempio la Op 17 che cambia la ripartizione e le Op 20 - 39 che funzionano correttamente.

A questo punto risulta sempre più misterioso perché nel Pgm 02 ci siano delle "stupide" sequenze del tipo 1 SUM 01 o anche peggio e delle quali si è già parlato nei numeri precedenti di MC. Io una risposta ce l'ho: forse ancora non conoscevano le Op ...

Terminiamo questa puntata con un'altra segnalazione.

Nel modulo "Math Utilities" è presente una utilissima sequenza, che permette la stampa indiretta di codici alfanumerici: a partire dal passo 179 del programma MU-03 (Pgm 03) c'è la sequenza "4 STO 01 RCL Ind 00 Op Ind 01 Op 30 Dsz 1 182 Op 05 Op 00 RTN". In particolare il passo 182 è quello contenente RCL Ind, mentre si può notare la presenza (finalmente!) di Op 30 per decrementare il contenuto del registro R00 (invece della sequenza 1 INV SUM 00).

Non dimentichiamoci che il modulo Math-utilities è costellato di HIR e, se la memoria non ci inganna, presenta anche il "Dsz esteso". Considerato che tale modulo è senz'altro più recente del "Master Library" in dotazione alle TI, forse tutto sommato ha ragione il nostro lettore nella sua ultima affermazione!

P.P.

invece di un INV SBR ci sia un R/S di fine elaborazione: in tal caso il registro di ritorno delle subroutine rimarrà riempito da un valore praticamente inutilizzabile in seguito. In particolare, se ci sono più situazioni del tipo "... A ... Lbl A ... R/S ..." può capitare (dopo 6 chiamate a pseudo-subroutine, dato che 6 sono i livelli di sotto-programmi permessi) che una chiamata ad una vera subroutine crei un'apparen-

te malfunzionamento.

È questo il caso del programma presentato, tanto è vero che nella lettera, Daniele Tincani imputava ad un funzionamento imperfetto del registro RTN della sua TI-58 alcuni errori che lo costringevano ad un certo punto all'uso di un RST, che tra le altre funzioni ha il compito di azzerare la catasta dei 6 registri "incriminati".

Perciò raccomandiamo di usare chiama-

te del tipo "... A ...", solo se si tratta di una subroutine e viceversa di usare la sequenza "... GTO A ..." (anche se apparentemente più "brutta") negli altri casi.

Tanto vale allora usare altri nomi per le etichette (a scelta tra circa 70 possibilità), lasciando alle 10 etichette A-E, A'-E' il loro compito fondamentale di far partire l'elaborazione, fatto questo peraltro ben noto ed usato un'infinità di volte. 