a cura di Pierluigi Panunzi

Torniamo in questo numero alla TI-57, dato che vi sono molti lettori a cui interessano programmi per questa calcolatrice. Parleremo, nel primo programma, della determinazione approssimata dell'orbita di un oggetto celeste, a partire da valori iniziali di velocità e delle coordinate, in base alle equazioni del moto. I dati forniti dal programma potranno essere riportati su di un foglio di carta millimetrata e si otterrà dunque l'andamento dell'orbita approssimata.

Il secondo programma serve, invece, a trasformare la TI-57 in orologio.

Calcolo coordinate dell'orbita di un satellite

di Stefano Chiti-Batelli
e Gabriele Toscani De Col - Roma

Il nostro programma calcola, in successive posizioni, l'orbita di un satellite. La memoria 0 e la memoria 5 servono al controllo del numero dei cicli di calcolo (vedi oltre). Nelle memorie 1, 2, 3, 4 sono rispettivamente l'ascissa x, la velocità Vx, l'ordinata y, la velocità Vy. La memoria 6 serve per i calcoli e viene automaticamente azzerata ad ogni ciclo al passo 29. La memoria 7 infine contiene ε (incremento temporale). Vediamo ora un esempio di esecuzione.

Con riferimento ad un sistema di unità di misura nel quale il prodotto tra la costante di gravitazione universale G e la massa del sole M è unitario (GM = 1) - questo sistema di u.d.m. è caratterizzato dal solo fatto di avere una diversa unità per le lunghezze che chiameremo ipermetro (Ipm): 1 Ipm = 5120 Km - impostiamo i seguenti dati iniziali:

x_0 = 0,5 Ipm
Vx_0 = 0,1 Ipm/s
y_0 = 0,0 Ipm
Vy_0 = 1,0 Ipm/s
ε = 0,05 s

ciascuno nella rispettiva memoria. Scriviamo ora 2 sul display (numero di cicli che si vogliono far effettuare alla macchina prima di conoscere le nuove x ed y; d'ora in poi per modificare tale numero occorrerà intervenire direttamente sulla memoria 5 inserendovi il nuovo numero ad esempio durante la visualizzazione di una coordinata prima, ovviamente, di premere R/S per proseguire l'elaborazione). A questo punto, dopo aver resettato (importante) la macchina, siamo pronti per l'elaborazione: premiamo R/S e dopo qualche secondo apparirà l'ascissa sul display. Premendo di nuovo R/S la macchina accenderà il valore della ordinata. Coordinate successive si ottengono premendo ancora R/S. Volendo conoscere anche i valori delle velocità dovremmo richiamare le memorie 2 e 4 rispettivamente per Vx e Vy.

Alcune considerazioni di carattere matematico per chiarire il metodo da noi seguito nella stesura del programma.

Siamo partiti dalla nota formula della gravitazione universale di Newton

$$\vec{F} = -G\,\frac{m\cdot M}{r^2}\,\frac{\vec{r}}{r}$$

che nel nostro sistema ipermetrico si scrive

$$\vec{F} = -\frac{m}{r^2}\,\frac{\vec{r}}{r}.$$

Tenendo presente la $\vec{F} = m\vec{a}$, segue:

$$\vec{a} = -\frac{\vec{r}}{r^3}.$$

Passando dal vettore alle sue coordinate abbiamo

$$a_x = -\frac{x}{(x^2+y^2)^{3/2}} \qquad \text{(passi da 09 a 25)}$$

$$a_y = -\frac{x}{(x^2+y^2)^{3/2}} \qquad \text{(passi da 28 a 34)}$$

Pertanto, nota la posizione del satellite, siamo in grado di ricavarci la sua accelerazione.

Nota l'accelerazione ricaviamo nel seguente modo la velocità:

$Vx(t+\varepsilon) = Vx(t) + \varepsilon a_x(t)$ (passi da 26 a 27)
$Vy(t+\varepsilon) = Vy(t) + \varepsilon a_y(t)$ (passi da 33 a 34)

Calcolo coordinate dell'orbita di un satellite

```
00   32 0        STO 0 25   85          =
01   32 5        STO 5 26   61 0      SBR 0
02   86 1        LBL 1 27   34 2      SUM 2
03   33 2        RCL 2 28   33 6      RCL 6
04   61 0        SBR 0 29  -34 6    I SUM 6
05   34 1        SUM 1 30   55          ×
06   33 4        RCL 4 31   33 3      RCL 3
07   61 0        SBR 0 32   85          =
08   34 3        SUM 3 33   61 0      SBR 0
09   33 1        RCL 1 34   34 4      SUM 4
10   23          X²    35   56        DSZ
11   34 6        SUM 6 36   51 1      GTO 1
12   33 3        RCL 3 37   33 1      RCL 1
13   23          X²    38   81        R/S
14   34 6        SUM 6 39   33 3      RCL 3
15   33 6        RCL 6 40   81        R/S
16   24          √X    41   33 5      RCL 5
17   35          Y^X   42   32 0      STO 0
18   03            3   43   51 1      GTO 1
19   85            =   44   86 0      LBL 0
20   25          1/X   45   55          ×
21   84          +/-   46   33 7      RCL 7
22   32 6        STO 6 47   85          =
23   55            ×   48  -61      I SBR
24   33 1        RCL 1
```

Orologio

```
00   32 1        STO 1 25   33 1      RCL 1
01   48 2        FIX 2 26   49        INT
02   61 3        SBR 3 27   32 1      STO 1
03   86 1        LBL 1 28   01          1
04   33 1        RCL 1 29   34 1      SUM 1
05   49          INT   30   33 1      RCL 1
06   32 7        STO 7 31  -76      I GE
07   83            .   32   61 3      SBR 3
08   06            6   33  -76      I GE
09   34 7        SUM 7 34   51 1      GTO 1
10   86 2        LBL 2 35   00          0
11   00            0   36   32 1      STO 1
12  -18        I LOG   37   61 3      SBR 3
13   83            .   38   51 1      GTO 1
14   00            0   39   86 3      LBL 3
15   01            1   40   03          3
16   34 1        SUM 1 41   00          0
17   33 1        RCL 1 42   32 0      STO 0
18  -76        I GE    43   33 1      RCL 1
19   61 3        SBR 3 44   86 5      LBL 5
20  -76        I GE    45   36        PAU
21   51 2        GTO 2 46   56        DSZ
22   02            2   47   51 5      GTO 5
23   04            4   48  -61      I SBR
24   32 7        STO 7
```

Analogamente, note le componenti della velocità, ci ricaviamo le coordinate del satellite:

$x(t+\varepsilon)=x(t)+\varepsilon Vx(t)$ (passi da 03 a 05)
$y(t+\varepsilon)=y(t)+\varepsilon Vy(t)$ (passi da 06 a 08).

Come avrete notato si tratta di uno sviluppo in serie di Taylor troncato per ovvi motivi al primo termine.

Il metodo seguito presenta tuttavia l'inconveniente che l'accuratezza con la quale viene calcolata l'orbita, ovverosia la sua ellitticità, dipende da questi due fattori:

— scelta di ε: tanto più piccolo ε tanto migliore la precisione nella determinazione dell'orbita;

— vicinanza al pianeta: dipendendo l'accelerazione da $1/r$, quando il corpo è molto vicino (r piccolo), visto il metodo di calcolo usato, se ε non è sufficientemente piccolo, la precisione cade. Es: con $r< 0{,}08$ Ipm suggeriamo $\varepsilon \leqslant 0{,}001$.

I tempi di calcolo risultano ovviamente collegati a tale discorso: essi saranno inversamente proporzionali alla scelta di ε (ε grande → tempi brevi, scarsa precisione; ε piccolo ...).

Con riguardo a ciò abbiamo infine incluso la possibilità di scegliere il numero di volte che si vuole la macchina calcoli le nuove coordinate senza presentare i risultati intermedi onde evitare la registrazione in una tabella dati, poi traducibile in grafico, di coordinate che, specie con ε molto piccolo, differiscano in modo irrilevante dalle precedenti trovate.

Un ultimo chiarimento: la subroutine 0 serve sostanzialmente a risparmiare passi (come sempre del resto) dovendosi eseguire quattro volte in uno stesso ciclo il prodotto per ε.

Orologio

di G. Domenico Facchetti - Treviglio (BG)

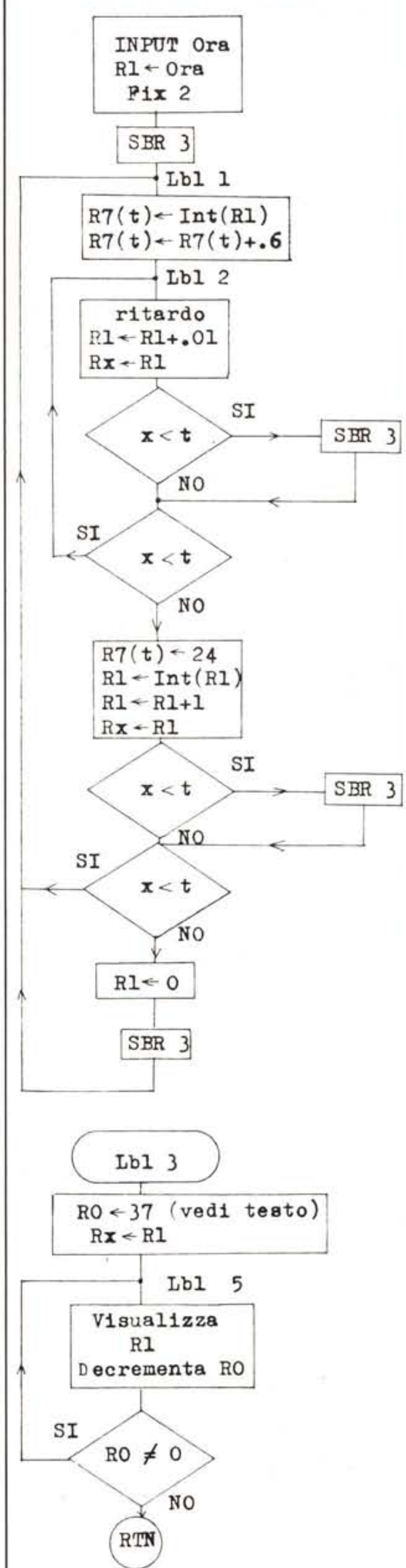
Il secondo programma permette invece alla calcolatrice di svolgere la funzione di un orologio, con visualizzazione quasi continua dell'orario in ore e minuti, separati dal punto decimale. Viene visualizzata l'ora dalle 0.00 alle 23.59, chiaramente con azzeramento automatico dopo le 24 ore. La precisione dell'ora segnata è discreta, considerato il mezzo usato, essendo l'errore massimo giornaliero di circa 5 secondi; la massima esattezza possibile si ha collegando la calcolatrice alla rete elettrica tramite l'apposito adattatore.

Il programma come si può vedere dal flow-chart è molto semplice: in particolare i passi 11 e 12 contengono un'operazione "inutile", che ha il solo scopo di dare al "minuto" la lunghezza più esatta possibile.

Il minuto viene fatto trascorrere dalla subroutine etichettata con Lbl 3, costituita da una successione di 37 "Pause" all'interno di un loop che viene eseguito per l'appunto 37 volte. A questo proposito, può capitare che differenti modelli di TI-57 abbiano una "velocità d'esecuzione" leggermente differente ed allora può essere richiesta una correzione di tale valore nei passi 40 e 41: ad esempio nel programma inviatoci, tale valore era 30.

Per eseguire il programma bisogna premere RST, impostare l'orario corrispondente al minuto che sta per scoccare (HH.MM) ed infine premere R/S, rilasciandolo solo quando sta scoccando il minuto desiderato. Nel caso che l'orologio-57 anticipasse o ritardasse bisogna, come visto, aumentare o diminuire (rispettivamente) il valore posto nei passi 40 e 41, effettuando un paio di tentativi.

Flow-chart programma Orologio



L'angolo delle TI

Alcuni lettori, avendo letto nei numeri precedenti dell'esistenza dell'istruzione HIR nelle TI-58-58C-59 e non possedendo il n° 4 di MCmicrocomputer (al quale rimandiamo per maggiori dettagli), ci hanno chiesto di ripresentarla. Accettiamo di buon grado, anche perché così saranno sempre di più gli utenti delle TI a conoscenza di tale "segreto di Pulcinella". L'istruzione HIR, del tutto assente nei manuali della TEXAS, è costituita dal codice 82, non corrispondente ad alcun tasto della tastiera e perciò impostabile in memoria solo con un piccolo artificio. È un'istruzione a due byte, il cui secondo byte ne specifica il funzionamento: a seconda del valore XY di tale secondo byte si avranno differenti operazioni. Prima di scendere in dettaglio, ricordiamo che le TI posseggono uno "stack" formato da 8 registri, utilizzato durante le normali operazioni matematiche, soprattutto quando vi sono calcoli in sospeso. Gli ultimi 4 di questi registri sono inoltre usati come buffer per la stampa ed infine sono usati da funzioni quali DMS,P/R e le funzioni statistiche. Ora il valore di Y del secondo byte della HIR si riferisce appunto ad uno di tali registri ($1 \leqslant Y \leqslant 8$) mentre il valore della X specifica qual è l'operazione da compiere sul registro prescelto. Si ha che:

Il valore di X	corrisponde a
0	STO
1	RCL
2	Nop
3	SUM
4	Prd
5	INV SUM
6,7,8,9	INV Prd

Ad esempio, per scrivere un dato nel registro 7 useremo HIR 07 o viceversa, per leggerne il contenuto, HIR 17. Per introdurre il codice 82 in un programma si può usare il semplice artificio di impostare un'istruzione del tipo STO 82 ed eliminando il byte relativo allo STO (codice 42), tramite l'istruzione Del, lasciando così "da solo" l'82.

Analogamente si opera per il secondo byte XY: oppure se tale codice corrispondesse ad un tasto effettivo si può premere tale tasto: ad esempio se XY = 33, invece di impostare STO 33 ed eliminare lo STO, si può più facilmente premere il tasto "x^2", il cui codice è per l'appunto 33.

P.P.