

Torniamo ancora una volta ad occuparci della TI-57, per la quale pubblichiamo due programmi: il primo riguarda la conversione di un numero da decimale ad esadecimale e sfrutta le nuove capacità "letterarie" della TI in questione; il secondo programma è invece dedicato a quei lettori che desiderano la pubblicazione di programmi di giochi e riguarda per l'appunto il "gioco dell'inversione".

Cogliamo al volo l'occasione per pregare (è proprio il verbo adatto) i lettori di non inviare più programmi riguardanti le conversioni: siamo letteralmente sommersi da programmi riguardanti questo argomento! Tra l'altro abbiamo scelto il programma proposto, sia per le sue doti di "novità", sia proprio per aprire e chiudere contemporaneamente l'argomento....

Ma sì, dato che ci siamo: cercate pure (è ancora con tono supplichevole che ve lo diciamo) di evitare programmi di bioritmi (ce n'è pure una versione sul manuale delle TI 58 e 59, che tutti si sentono in dovere di migliorare!) e soluzioni di equazioni (a meno che non scoviate qualche metodo strano, oltre a quelli soliti di Newton, delle parabole, delle tangenti, delle secanti, e chi più ne ha più ne metta).

Terminato questo sfogo veniamo ai due programmi.

Conversione decimale-esadecimale

di Daniele Bolognari (Padova)

Dato che la maggior parte dei lettori conosce senza dubbio l'algoritmo che permette la conversione di un numero da decimale ad esadecimale (ed anche chi non lo conoscesse ancora, può ricavarlo semplicemente dal listato del programma), ricordiamo soltanto che le cifre esadecimali vanno da "0" ad "F" ed in particolare le ultime sei (da "a" ad "f") potrebbero porre problemi per la visualizzazione.

Oramai sappiamo che la nostra TI-57 è in grado di visualizzarle, grazie al "metodo della sequenza generatrice" segnalato nell'"Angolo delle TI" del n. 8 di MCmicrocomputer, per cui anche questo problema è risolto.

Il programma di conversioni risulta molto semplice concettualmente: ricevuto in input un numero N maggiore di 0, intero e minore di 99999999 (altrimenti si potrebbero verificare errori nella conversione), a seguito dell'elaborazione verranno visualizzate volta per volta le cifre che compongono il numero esadecimale corrispondente ad N, a partire da quella meno significativa, fino a quella più significativa.

Perciò, per rappresentare correttamente sul display una cifra esadecimale, viene effettuato un confronto: se tale cifra è minore di 10 essa viene visualizzata (per circa un secondo) tramite un'istruzione di Pause; se invece tale cifra è maggiore di 10, ma sempre minore di 16, viene sottratto ad essa il valore 10 ed il numero così ottenuto fa saltare, tramite un SBR indiretto, alla corrispondente parte di programma che permette appunto la visualizzazione di una lettera.

Ognuna di queste parti contiene un codice "artificiale" (corrispondente per l'appunto ad una lettera esadecimale) e subito dopo un RST che rimanda al "Pause" del passo 00. Questo perché RST è l'unica istruzione di salto incondizionato che permetta la visualizzazione delle lettere: tra l'altro questo fatto permette di risparmiare passi di programma.

Per quanto riguarda la memorizzazione del programma, ed in particolare dei passi "artificiali", consigliamo di seguire la seguente procedura: memorizzare la "sequenza generatrice" a partire dal passo 20, generare prima la "f", poi la "e", ecc; fino alla "a", introducendo ogni volta un passo di programma vuoto, tramite un "Ins".

Fatto ciò, possiamo introdurre il programma a partire dal passo 00 e fino al passo 22: a questo punto si introduce il SBR indiretto premendo "SBR SST". Quindi, con molta attenzione (per non cancellare i codici artificiali già introdotti), e tramite le istruzioni "Ins", "Del", "SST" e "BST", bisogna impostare i successivi passi del programma, fino a 49.

Ecco che abbiamo perciò memorizzato (con non poca fatica!) il nostro programma; ora bisogna porre il valore 16 in STO 0 ed il valore 10 in STO 1: con ciò siamo veramente pronti.

Impostiamo un numero N, ad esempio proprio il valore massimo consentito e cioè 99999999 e premiamo SBR 4: volta per volta verranno visualizzate le cifre esadecimali a partire da quella meno significativa.

In brevissimo tempo (a proposito: bisogna essere un tantino veloci per scrivere le cifre presentate via via sul display: se ne perdiamo qualcuna tanto vale fermare l'elaborazione con R/S e ricominciare reimpostando il numero N e premendo SBR 4) dicevamo, nel nostro caso otterremo "5F5EdFF", che, manco a farlo apposta, è proprio il valore esatto!... Funziona....

Conversione decimale - esadecimale

00	36	PaU	25	10	"A"
01	86 0	Lb1 0	26	71	RST
02	33 0	RCL 0	27	01	1
03	- 39 2	I Prd 2	28	15	CLR
04	33 2	RCL 2	29	11	"b"
05	19	C.t	30	71	RST
06	66	EQ	31	02	2
07	81	R/S	32	15	CLR
08	- 49	I Int	33	12	"C"
09	- 34 2	I SUM 2	34	71	RST
10	32 3	STO 3	35	03	3
11	33 1	RCL 1	36	15	CLR
12	32 7	STO 7	37	13	"d"
13	33 0	RCL 0	38	71	RST
14	39 3	Prd 3	39	04	4
15	33 3	RCL 3	40	15	CLR
16	- 76	I GE	41	14	"E"
17	71	RST	42	71	RST
18	65	-	43	05	5
19	33 1	RCL 1	44	15	CLR
20	85	=	45	15	"F"
21	- 18	I log	46	71	RST
22	61	SBR (SST)	47	86 4	Lb1 4
		0	48	32 2	STO 2
23	00		49	51 0	GT0 0
24	15	CLR			

Reverse

di Sandro Cosenza (Verona)

Si tratta di un solitario e senza dubbio è uno dei pochissimi giochi non basati sulla fortuna che mi sia mai trovato ad affrontare, almeno con la mia umile TI-57...

(N.d.r. Già sul numero 15 è stato pubblicato, nella rubrica Software R.P.N., un programma analogo per HP-41: lungi dall'essere una ripetizione, il programma che presentiamo dimostra che anche con la TI-57 si può "fare qualcosa"... Ovviamente ci saranno alcune limitazioni: ad esempio non darà delle classifiche, anche perché nel nostro caso si tratta di un solitario....).

L'obiettivo è quello di disporre le 8 cifre dall'1 all'8, che all'inizio sono messe a caso, nel loro ordine naturale, mediante l'inversione delle prime N cifre di destra.

Faccio un esempio: se sul display ho 12347865, ponendo N=4, avrò 12345687;

in seguito, usando 2, otterrò 12345678, terminando la partita.

La differenza fra questo ed altri giochi simili, però, è che qui si può vincere o perdere. Infatti, per scombicare il numero base (12345678) la calcolatrice si crea un N, tante volte quanto è il numero imposta-

to inizialmente dal giocatore (non il numero che serve direttamente per fare le inversioni....), se si vuole vincere, così, bisogna ritornare al numero base entro e non oltre il numero di inversioni operate dalla calcolatrice.

Non solo! Proprio perché ognuna di

"Reverse"				16	38	2	Exc	2	33	56	Dsz			
00	32	4	STO	4	17	66	EQ		34	51	2	GTO	2	
01	42		EE		18	51	0	GTO	0	35	22	x	t	
02	86	0	Lbl	0	19	01		1		36	-	18	I log	
03	33	1	RCL	1	20	-	34	4	I SUM	4	37	39	5	Prd
04	-	18	I log		21	86	2	Lbl	2	38	33	5	RCL	5
05	-	49	I Int		22	-	18	I log		39	34	3	SUM	3
06	32	1	STO	1	23	-	39	5	I Prd	5	40	00		0
07	55		x		24	33	5	RCL	5	41	38	3	Exc	3
08	07		7		25	-	49	I Int		42	32	5	STO	5
09	75		+		26	-	34	5	I SUM	5	43	33	4	RCL
10	02		2		27	55		x		44	76		GE	
11	85		=		28	33	0	RCL	0	45	51	0	GTO	0
12	49		Int		29	-	18	I log		46	33	5	RCL	5
13	86	1	Lbl	1	30	85		=		47	-	42	I EE	
14	32	0	STO	0	31	34	3	SUM	3	48	81		R/S	
15	32	7	STO	7	32	01		1		49	51	1	GTO	1

L'ANGOLO DELLE TI

Al lettore attento, ed in particolare ai "S.O.A.-isti", non sarà sfuggita l'importanza degli argomenti via via riportati nell'"Angolo delle TI" a partire dal n. 2 di MCmicrocomputer: alcuni erano fondamentalmente dettati dall'esperienza nell'uso delle calcolatrici programmabili della Texas Instruments, ben al di là di quanto segnalato nei pur ottimi manuali d'uso e di riferimento, altri argomenti invece esulavano completamente dalla sfera della normalità d'uso, sconfinando al di là di barriere oramai ampiamente sgretolate...

Inutile dire che ci stiamo riferendo al "gioiello" della nostra collezione di argomenti, di cui abbiamo parlato lo scorso numero: la generazione di una montagna di nuove istruzioni, aventi il codice espresso in notazione esadecimale.

Davanti a questa scoperta, del tutto inattesa (parola di "veterano cinquantanovista"), l'oramai vecchiotta istruzione HIR diventa addirittura antidiluviana: parlare di "synthetic programming" in questo caso (a buon intenditor....) è del tutto inesatto in quanto il codice 82 (HIR per l'appunto) è lì pronto per essere trovato da chiunque.

Ecco che invece la generazione dei codici esadecimali può a diritto essere considerata "sintetica" (inglesismi a parte!) dal momento che richiede l'uso di una sequenza che almeno funzionalmente ricorda il "Byte Jumper" di HP-quarantuniana memoria. Comunque, come i "mass media" ci insegnano, diffidate delle imitazioni!!!!.....

A beneficio dei lettori ricordiamo che con la "Sequenza di Laporta" (bello, eh?!) è possibile generare qualunque codice esadecimale (badate bene!) compreso tra 00 e 9f: i valori "decimale", ad esempio 23, 41, 54, ecc., non sono altro che i consueti codici normali, mentre viceversa i valori propriamente esadecimali, ad esempio 2a, 5f, 9c, ecc., si riferiscono a funzioni del tutto nuove.

Parlando genericamente, in quanto ci riserviamo di ritornare più in dettaglio sull'argomento già dal prossimo numero, le funzioni ottenibili sono molto strane: alcune sono alquanto suscettibili e mandano in pallone la povera TI, altre hanno nomi (solo in "TRACE") non del tutto fantasiosi, altre ancora non sono listabili oppure fanno entrare la stampante in loop mentre la poverina doveva semplicemente listare un programma, contenente però un "elemento di disturbo".

Però vi sono anche funzioni più importanti, ad esempio

quella che permette la lettura di un registro interno alla calcolatrice (inutile dire che finora era del tutto invisibile!) contenente lo stato dei flag....

...fine della puntata....

Eh sì! Lo spazio a disposizione dell'"Angolo" non consente di parlare ancora delle funzioni "sintetiche": a risentirci sul prossimo numero.

* * *

Questa volta proponiamo ai lettori il contributo di Sandro Cosenza, del quale abbiamo anche pubblicato un programma in questo stesso numero. Vediamo cosa ci dice e passiamogli la parola.

Riguardo all'"Angolo delle TI", del numero 8, che parlava della creazione di codici alfabetici e strane istruzioni sulla TI-57, vorrei continuare il discorso.

L'articolo terminava con l'invito a cercare nuovi "buchi" nel sistema operativo, magari scoprendo nuove istruzioni relative ai codici 16, 17, 37, 47 e a quelli corrispondenti alla posizione dei tasti numerici.

Per questi ultimi.... beh, niente da fare (n.d.r. o no?!): quando ci si aspetta che vengano fuori, escono sempre altri codici composti; riguardo ai primi 4 codici citati, li ho trovati tutti (anzi, per la precisione ho trovato i loro "inversi", ma tanto, a quanto pare, in questi casi dovrebbe essere la stessa cosa), ma purtroppo non sono nulla di nuovo...

Il primo, il codice 16, è ottenibile usando come sequenza generatrice 3:3:3 e "0" come numero da impostare successivamente; il suo funzionamento è uguale a quello del codice "11 non esadecimale".

I codici 47, 37, e 17 vengono generati dalla sequenza 3:3:3: +/- e dai numeri, rispettivamente, 3, 2 e 0: i primi due sono uguali, come funzioni esplicabili, all'istruzione "Nop"; il "17" è del tutto simile al "12 non esadecimale", ma riguardo a quest'ultimo ci sono delle novità.

Provate infatti a memorizzare il programma "12" R/S RST dopodiché uscite dal modo LRN e premete 10 RST R/S R/S +/- . +/-

Se ne vedranno delle belle! Provare per credere!

Ancora più belle, però, si potranno vedere operando in modo "interruttore software", visto nel n. 9 di MCmicrocomputer.

queste si basa su di un N a caso, potrebbe capitare di usarne due uguali consecutivi, ritornando alla situazione antecedente la loro creazione e incrementando inutilmente il contatore (R₄). Viene allora operato un confronto fra ogni N e il proprio precedente: se i due valori sono uguali, ne viene creato un altro, e così via; bisogna quindi porre attenzione. Il test è fatto anche fra l'ultimo N creato dalla calcolatrice e il primo impostato da tastiera, nonché fra i vari N usati dal giocatore!

Quindi: occhio all'ultimo tempo di elaborazione prima dell'inizio attivo del gioco e occhio a non sbagliare N e poi volere tornare indietro (con una sola mossa): il tutto verrebbe scombinato ancora peggio. (Non escludo che ci saranno anche quei fortunati ai quali la situazione, in questo modo, verrebbe migliorata: ma io non ci conterei molto....)

Come si può vedere, quindi, il gioco è arricchito di difficoltà, che, a mio parere, non guastano proprio.

(Sarebbe stato più comodo avere anche la visualizzazione automatica del numero di inversioni operate per risolvere il gioco, ma purtroppo, la memoria della TI 57 è quella che è).

Esecuzione

Dopo aver caricato il programma:

- a) cancellare tutte le memorie con INV C.t;
- b) memorizzare 12345678 in R₅
- c) memorizzare un numero frazionario a caso non apparente e minore di 10 in R₁;
- d) fissare i decimali a ZERO; (cioè, insieme all'entrata in notazione esponenziale (passo 01), simula la "C" durante l'elaborazione delle 58, 58-C, 59, dato che con l'esperienza, a causa della lentezza della 57, si può arrivare a seguire la generazione del numero base scombinato).
- e) impostare il numero di inversioni che si vogliono far operare dalla calcolatrice per scombinare il numero base (io consiglio di rimanere nei limiti da 5 a 10)
- f) digitare RST e R/S e attendere che la

"lunghezza" elaborazione termini; (ora incomincia il gioco vero e proprio)

g) impostare N e premere R/S; questo fino a che non si sarà arrivati a 12345678;

h) a questo punto digitare RCL 4, per leggere il numero di inversioni operate per risolvere il gioco: come già visto, se il suo valore assoluto (il numero è negativo) è minore o uguale a quello impostato nel punto "e", si ha vinto; al contrario, ci si sarà lasciati "battere" da una misera 57....

Per giocare ancora, partire dal punto "e".

Concludiamo segnalando l'occupazione dei vari registri:

R0	numero trasformatore (generato da programma)
R1	numero generatore
R2	numero trasformatore precedente
R3	numero trasformato
R4	contatore
R5	numero da trasformare (12345678)
R6	non usata
R7	registro "t" per i confronti

Dopo aver impostato INV STO 3 ("spegnendo" così la TI-57), premete

CLR RST R/S. 111 +/- R/S +/- INV INV

(no, non è un errore! bisogna digitare due volte il tasto INV); infine premete EE 1

Che si sia impazzita la nostra povera TI?

Infine, dato che siamo in vena di torture, propongo di memorizzare in programma due qualsiasi "lettere esadecimali" (sempre quelle del n. 8 di MC), di farle seguire dalle due istruzioni R/S RST e di eliminare con l'istruzione "Del" il programma generatore (che in questo caso è "Exc Lbl 1 3 = R/S").

In tal modo, a partire dal passo 00 avremo i seguenti codici: 13 10 81 71 dove "13" e "10" corrispondono alle lettere "d" ed "A", scelte per l'appunto a caso.

Ora usciti dal modo LRN, premiamo "RST EE R/S R/S .1".

Già si vede subito che c'è qualcosa di strano: sul display si leggerà: 0. - dA

Ora, digitando per sei volte la sequenza "EE R/S .1" si introdurranno nel display altri segni "-", come se fossero delle normali cifre decimali! Ma non è finita! Premiamo ancora "+/- EE +/-". A questo punto otteniamo -0. ----- dA

Simpatico, no?! Tra l'altro, giocando sul numero di volte che si preme R/S, si otterranno leggere variazioni sul tema.

Robe da accumulatori scarichi!...

A conclusione di questo "angolo delle TI" rinnoviamo l'invito già precedentemente rivolto ai lettori di collaborare con idee, spunti, soprattutto ora che ci si presenta l'opportunità di scoprire il funzionamento di nuove istruzioni, sulle cui caratteristiche ritorneremo più volte.

L'indice dell'Angolo

Vogliamo far cosa gradita a quei lettori che non hanno seguito tutte le puntate di questa rubricetta, riportando un elenco degli argomenti trattati nei vari numeri di MCmicrocomputer nell'angolo delle TI.

- 1) L'"Angolo delle TI" non era ancora nato....
- 2) Considerazioni sui tempi di elaborazione con particolare riferimento agli indirizzamenti assoluti e tramite etichette.
- 3) Analisi dettagliata delle istruzioni "Op nn", con aggiunte dettate dall'esperienza d'uso.
- 4) L'istruzione HIR? una funzione utile e "dimenticata" dai manuali, ma scoperta alcuni anni fa dai "texasiani degli States" e utilizzata nei moduli Solid State Software più recenti.
- 5) Caratteristiche della TI-57, con particolare riferimento alle differenze con i modelli maggiori.
- 6) L'istruzione Dsz "estesa" ed una ventina di etichette nuove.
- 7) Prova dei flag da tastiera, stranezze (le prime di una lunga serie...) nel funzionamento delle TI, registri HIR usati da alcune funzioni.
- 8) Nuove istruzioni sulla TI-57: come far apparire le

lettere "abcdef" sul display, l'istruzione SBR Indiretta ed altro.

9) Come spegnere il display della TI-57, senza usare l'interruttore della calcolatrice, ma via software.

10) Alla scoperta del sistema operativo delle TI-58 e 59: la ROM delle istruzioni "complesse" (statistiche e di conversione).

11) Alcune piccole utility all'interno dei programmi del modulo di libreria "master library".

12) Ancora sull'istruzione HIR.

13) Analisi della ROM delle funzioni, simboli strani (gradi, primi e secondi) sul display delle TI-58 e 59 ed alcuni metodi per "torturare" la propria calcolatrice facendole eseguire programmi ben strani.

14) Ancora sulla ROM ed altre stranezze (512 byte con la TI-58). Lo spegnimento del display della TI-57 da programma.

15) Sintesi "non-standard" di due nuove funzioni: una di queste si chiama "TLR"

16) La "Programmazione sintetica", ovvero come generare qualsiasi istruzione avente codice esadecimale: superate le frontiere del "decimale" con 60 nuove istruzioni (si sono proprio sessanta!).