

Routine generazione numeri casuali non ripetitivi

di Enrico Pierucci - Genova

La routine che propongo è nata dalla necessità di generare una serie di numeri casuali senza mai ripetere quelli già generati (l'esempio più classico è la tombola).

Il problema da risolvere era quello dello scarto dei numeri non più accettabili, che, nella sua soluzione più semplice, prevede la loro cancellazione da una tabella preparata allo scopo mano a mano che vengono generati, con rieffettuazione del ciclo di generazione se si riscontra, che il numero è già stato cancellato. Questo sistema però comporta tempi di attesa via via crescenti e non accettabili nella fase finale, quando i numeri da scartare superano abbondantemente quelli buoni.

Ho pensato allora ad un algoritmo che permette di mantenere inalterato il tempo di esecuzione dalla prima all'ultima generazione essendo completamente slegato dalla quantità già generata o ancora da generare.

La soluzione consiste nel creare sempre la tabella con tutti i numeri che si vogliono generare, immettendo il fattore casuale nell'indice di puntamento alla tabella stessa. Alla successiva generazione sarà sufficiente ridurre di una unità l'ampiezza della tabella perdendo l'ultimo elemento che deve essere sostituito in quello appena generato.

Questo sistema, oltre a risolvere il problema di partenza, si presenta molto duttile potendo generare sia numeri che stringhe alfanumeriche, essendo l'output determinato dall'iniziale carico della tabella.

La routine utilizza il registro 00, la tabella va caricata a partire dal registro 01, e la routine di generazione di un numero casuale

da 0 a 1 è stata ripresa dal libro di applicazione standard HP41C.

Ottimo lavoro! Il programma del Sig. Pierucci è decisamente geniale e funziona in modo impeccabile. Forse non altrettanto brillante è la descrizione fatta dall'autore sul suo funzionamento. Il programma utilizza, per ricordare i numeri già usciti, una serie di tanti registri quanti sono gli elementi da estrarre; all'inizio della serie di sorteggi, ogni elemento va caricato in un registro, a partire dal registro R01 e finendo al registro Rnn, dove nn è il numero di elementi da sorteggiare. Parlo di "elementi" e non di "numeri", poiché se noi volessimo sorteggiare ad esempio dei nomi, basterebbe semplicemente caricarli nei registri al posto dei numeri poiché il programma si comporta proprio come una mano che va a pescare nel sacchetto del sorteggio, costituito nel nostro caso dalle memorie dati. Vediamo più dettagliatamente come lavora il programma. Inizialmente tutti gli elementi da estrarre a sorte si trovano nei registri dati da R01 a Rnn; allorché viene chiesto il primo sorteggio, un generatore di numeri pseudocasuali fornisce un numero casuale NN compreso tra 1 e nn inclusi, detto numero indicherà al puntatore di estrarre e visualizzare il contenuto del registro RNN cioè l'NN-esimo elemento della tabella; questo, una volta visualizzato, non occorre più e il programma provvederà allora a sostituirlo con l'nn-esimo elemento, cioè richiama il contenuto del registro Rnn (l'ultimo della tabella) e lo memorizza in RNN; dopo di ciò, viene ridotto di una unità il limite superiore dei numeri casuali forniti dal generatore, escludendo automaticamente l'ultimo elemento della tabella che però ora si trova memorizzato

in RNN al posto dell'elemento già estratto. Alla successiva richiesta, il procedimento si ripete ma con la tabella ridotta di uno, e così via fino all'esaurimento degli elementi del sorteggio, annunciato da un BEEP. Un neo del programma proposto dall'autore, è costituito dal fatto di dover memorizzare manualmente la tabella degli elementi del sorteggio, ma nulla impedisce di utilizzare un altro programmino per caricare, con un semplice loop, tutti i registri occorrenti. Mi sono permesso quindi di modificare il programma originale introducendo una routine di inizializzazione che provvede automaticamente a caricare in memoria i numeri da estrarre; con la facoltà, da parte dell'operatore, di scegliere il limite massimo della serie di numeri da estrarre. Tale routine controlla automaticamente i dati in ingresso, prendendo soltanto la parte positiva intera del numero impostato; ovviamente il SIZE impostato dovrà essere superiore di una unità al numero di elementi da estrarre, pena la comparsa del messaggio "NONEXISTENT". Il programma si usa nel modo seguente:

- 1) Per inizializzare il programma, impostare XEQ "IN", alla richiesta "SEME" digitare un numero decimale compreso tra 0 e 1 esclusi e premere R/S.
- 2) Alla richiesta "NUMERO MAX?" impostare il numero più alto che si desidera sorteggiare e premere R/S, dopo alcuni secondi comparirà sul display il messaggio "-PRONTO-".
- 3) Per estrarre un numero, premere R/S
- 4) Dopo aver estratto l'ultimo numero, premendo ancora R/S, un BEEP annuncerà la fine del sorteggio
- 5) Per ricominciare con una nuova serie di estrazioni, premere XEQ "GEN" e riprendere dal punto 2.



Routine generazione numeri casuali non ripetitivi

01*LBL "IN"	11 ABS	21 STO IND L	31 *	41 BEEP
02 CF 29	12 I E3	22 ISG L	32 1	42 STOP
03 FIX 0	13 /	23 GTO 03	33 +	43*LBL 02
04 *SEME ?"	14 FRC	24 "-PRONTO-"	34 RCL IND X	44 RCL 00
05 PROMPT	15 1	25 PROMPT	35 STOP	45 9821
06 STO 00	16 +	26*LBL 01	36 RCL IND Z	46 *
07*LBL "GEN"	17 STO L	27 XEQ 02	37 STO IND Z	47 .211327
08 "NUMERO MAX. ?"	18*LBL 03	28 RCL Y	38 Rf	48 +
09 PROMPT	19 RCL L	29 1	39 DSE X	49 FRC
10 INT	20 INT	30 -	40 GTO 01	50 STO 00
				51 END

L'INPUT DEI DATI

Sempre, nella compilazione di un programma, va esaminato e curato il problema dell'input dei dati; sia che si tratti di un semplice programma che lavora su un solo dato, sia che ci si trovi di fronte a un complesso procedimento di elaborazione operante su una grossa quantità di dati. Purtroppo, assai spesso si trascura questo problema, ed è allora che vengono fuori dei programmi che è difficile usare, semplicemente perché non si sa come dar loro "in pasto" i dati da elaborare. Vediamo allora qualche possibile forma di input da utilizzare sulla 41C/CV. La forma di input più semplice è quella ottenuta tramite una istruzione di STOP nel punto del programma in cui va introdotto il dato: questa soluzione, pur funzionando, può dare adito a errori poiché nel caso in cui i dati da introdurre siano numerosi si può facilmente "perdere il filo" e scambiare tra loro le informazioni fornite al calcolatore, non mostrando quest'ultimo alcun segno di identificazione del dato da impostare in quel momento; perciò questa soluzione è senz'altro da evitare, a meno che si tratti di programmi brevissimi per i quali l'ordine di input è inequivocabile. Disponendo, la nostra 41, della possibilità di mostrare stringhe alfanumeriche sul display, è bene sempre identificare con un messaggio (possibilmente scritto in forma chiara e usando spaziature e abbreviazioni che conferiscono una certa omogeneità estetica ai vari messaggi) ciascun arresto del programma per l'introduzione di un dato. Il sistema migliore è di introdurre nel registro ALPHA il messaggio di richiesta e quindi far seguire l'istruzione "PROMPT". la giusta sequenza di istruzioni è la seguente:

```
.....
"DATO X?"      (messaggio di richiesta del dato)
PROMPT
.....
```

Nell'esempio in questione la calcolatrice si arresterà visualizzando la domanda "DATO X?", a questo punto basta impostare le cifre (che verranno introdotte nel registro X) e premere R/S per far riprendere l'elaborazione dall'istruzione immediatamente seguente il PROMPT. Notate che prima dell'istruzione "DATO X?" Non ho messo un CLA, dato che l'impostazione della nuova stringa cancella automaticamente quanto già si trovasse nel registro ALPHA.

Quando è necessario impostare dati costituiti da stringhe alfanumeriche, una buona sequenza da utilizzare è la seguente:

```
.....
AON (attiva il modo ALPHA)
"DATO ALPHA?" (messaggio di richiesta del dato)
PROMPT oppure STOP
AOFF (disattiva il modo ALPHA)
.....
```

Questa sequenza fa fermare la macchina con il modo ALPHA già attivato, per cui basta premere i tasti relativi ai caratteri da impostare e quindi R/S per proseguire l'elaborazione; il messaggio impostato sostituirà automaticamente il precedente contenuto del registro ALPHA. Talvolta è necessario poter saltare l'introduzione di un certo dato, per esempio nel caso in cui questo non serve o deve rimanere quello impostato in una precedente elaborazione. Una buona idea è quella di utilizzare il Flag 22, che si accende ogni volta che viene impostato un dato numerico, la procedura può essere questa:

```
.....
"DATO ?"
PROMPT
FS? C 22
STO 01
.....
```

Nell'esempio in questione ho supposto che il dato introdotto debba essere memorizzato nel registro R01; se non si imposta alcuna cifra ma si preme semplicemente R/S il Flag 22 non verrà attivato e quindi il puntatore salterà l'istruzione STO 01 lasciando inalterato il contenuto del registro stesso. È questa una prima forma di "controllo" dell'input: il calcolatore comincia a "ragionare" su quanto impostato, senza accettare ciecamente qualsiasi cosa in ingresso. Sovente si rende necessario controllare più restrittivamente gli input, allo scopo di non provocare errori a causa di dati rappresentanti informazioni sicuramente errate. Per esempio, è sicuramente errato fornire un numero decimale quale dato in input a routine per il calcolo dei fattoriali; sarà utile allora una sequenza che, subito dopo l'input, controlli che sia stato impostato un numero privo di decimali. Tale sequenza può comportarsi in vari modi nel caso in cui l'input risultasse illecito:

- 1) Chiedere di nuovo il dato.
 - 2) Visualizzare un messaggio di errore.
 - 3) Correggere automaticamente il dato impostato e proseguire nell'elaborazione.
- Per fare un esempio, vediamo come si può impostare una routine di controllo per l'input di un programma per il calcolo dei fattoriali. Prima di tutto bisogna definire quale sia l'insieme degli input leciti e, per esclusione, l'insieme di quelli illeciti; nel nostro caso sono leciti soltanto i numeri interi non negativi da 0 a 69.
- Ecco quindi tre routine per il calcolo dei fattoriali diverse tra loro soltanto nel tipo di controllo effettuato (ovviamente la routine per il calcolo dei fattoriali è riportata soltanto a scopo d'esempio, in quanto la funzione fattoriale è compresa già tra le standard della 41C).

La routine "FATT 1" controlla il dato impostato e, se l'input è illecito, ripete la

domanda finché non si introduce un dato lecito, il funzionamento è semplice, al passo 04 l'elaborazione si arresta e la 41 visualizza la domanda "NUMERO?" dopo aver impostato il numero e aver premuto R/S le istruzioni 05 e 06 introducono il numero 69 nel registro Y, il passo 07 $x \leq y?$ confronta quindi tale numero col dato impostato, se quest'ultimo è minore o uguale a 69 il programma procede col passo seguente, altrimenti salta direttamente al passo 09 GTO 01 (input illecito) e quindi di nuovo alla LBL 01 per ripetere la domanda "NUMERO?"; il passo 08 $x < 0?$ controlla che il numero impostato non sia negativo, qualora lo fosse, l'esecuzione passerebbe al seguente GTO 01 e quindi di nuovo alla LBL 01 per ripetere la richiesta di dati; ultimo controllo è la verifica dell'assenza di parte decimale, per fare ciò, i passi 10, 11 e 12 operano confronto tra il numero impostato e la sola parte intera di esso, se c'è differenza, l'esecuzione salta alla LBL 01 tramite il GTO 01, altrimenti anche questo test è stato superato e ciò vuol dire che il dato è lecito e si può passare al calcolo vero e proprio, affidato ai passi dal 14 in poi.

01*LBL "FATT1"	01*LBL "FATT2"	01*LBL "FATT3"
02*LBL 01	02 "NUMERO ?"	02 "NUMERO ?"
03 "NUMERO ?"	03 PROMPT	03 PROMPT
04 PROMPT	04 69	04 ABS
05 69	05 X<>Y	05 INT
06 X<>Y	06 X<=Y?	06 69
07 X<=Y?	07 X<0?	07 X>Y?
08 X<0?	08 GTO 01	08 RDN
09 GTO 01	09 ENTER↑	09 STO L
10 ENTER↑	10 INT	10 DSE L
11 INT	11 X*Y?	11*LBL 02
12 X*Y?	12 GTO 01	12 RCL L
13 GTO 01	13 STO L	13 *
14 STO L	14 DSE L	14 DSE L
15 DSE L	15*LBL 02	15 GTO 02
16*LBL 02	16 RCL L	16 X<=0?
17 RCL L	17 *	17 1
18 *	18 DSE L	18 RTN
19 DSE L	19 GTO 02	19 END
20 GTO 02	20 X<=0?	
21 X<=0?	21 1	
22 1	22 RTN	
23 RTN	23*LBL 01	
24 END	24 "ERROR"	
	25 TONE 8	
	26 AVIEW	
	27 END	

Pur non avendo molto a che fare con l'argomento che stiamo trattando, vale la pena di dare un'occhiata anche al metodo usato per il calcolo dei fattoriali, che fa uso della sola catasta senza neppure "scombussolarla" troppo. La routine "FATT2" è identica alla "FATT1" ma anziché inviare l'elaborazione di nuovo all'inizio, la condizione di "dato illecito" fa saltare l'elaborazione a una routine che dopo un messaggio sonoro (niente pernacchie, soltanto un bip) arresta l'esecuzione visualizzando "ERROR". La routine "FATT3", infine, accetta comunque il dato introdotto e, nel caso in cui questo fosse illecito, si limita a correggerlo; il procedimento è più semplice dei precedenti: dopo l'introduzione del dato, il passo 04 rende il numero introdotto sicuramente positivo, il passo 05 gli toglie una eventuale parte decimale; i passi 06, 07 e 08 verificano che l'input non sia superiore a 69 e se così fosse lo riducono a tale quantità; dopo questi "trattamenti" il numero è stato forzatamente reso lecito e quindi l'elaborazione può procedere. Questo terzo sistema di controllo è il più pericoloso dei tre, perché l'operatore può non accorgersi di aver introdotto dei dati illeciti e ottenere elaborazioni apparentemente corrette su dati che invece sono stati modificati dalla routine di controllo; quindi un simile tipo di procedimento è consigliabile soltanto per controllare dati illeciti che l'operatore avrebbe potuto volontariamente impostare, per esempio nel caso di giochi, dove talvolta si tenta di "ingannare" il calcolatore. Gli esempi che ho mostrato servono a dare una idea di come si può rendere la 41C capace di distinguere un input corretto da uno errato e inaccettabile, dando così quel tocco di "intelligenza" in più ad ogni programma. Scusandomi con chi, ritenendo scontate queste considerazioni, si sentirà preso in giro da queste righe e da quelle che scriverò ancora, spero di aver chiarito uno dei tanti problemi che si presentano nella programmazione del nostro caro "macinino".

P.G.

L'efficienza di un computer dipende dalle periferiche. L'efficienza delle periferiche dipende da SEGI.

SEGI, continuando a proporre alla sua clientela l'affidabilità e la tempestività dei propri servizi di assistenza, annuncia la disponibilità di due nuovi terminali video HAZELTINE, creati per rinnovare il design, il prezzo, le prestazioni.

EXECUTIVE 10 Il modello più competitivo della prestigiosa linea di video terminali HAZELTINE serie EXECUTIVE

- tastiera ergonomica a basso profilo
- schermo girevole/inclinabile
- completa capacità di editing
- 25 linee da 80 caratteri
- 8 tasti funzione, programmabili
- capacità grafica di tracciamento linee partizioni) e dei campi (in negativo, lampeggiante, alta/bassa intensità, protezione ecc.)
- interfaccia ausiliaria di I/O per collegamento con periferiche

ESPRIT II La classe nell'economicità del prezzo

- tastiera separata
- schermo antiriflesso, 12 pollici, a fosfori verdi
- 128 caratteri visualizzabili con matrice 7x11
- collegabilità con stampante hard-copy fino a 19200 baud
- inserimento e/o annullamento di caratteri
- ampia capacità di emulazione di altri video.



AMITALIA, SAICO, SEGI: tre leader.
un gruppo. AMMI.

...FIDATI DI SEGI È UN NOME CHE CONTA NELL'INFORMATICA

segi SERVIZI
GENERALI PER
L'INFORMATICA
SPA

SEGI - Via Timavo, 12 - 20124 Milano
Tel. (02) 6709136 (5 linee ricerca automatica) - Telex 315132 I
SEGI - Via Asmara, 58 - 00199 Roma
Tel. (06) 8395766 - Telex 616130 I