For presentation at the 25th meeting of the Parapsychological Association, Cambridge, England, August 1982.

A Radomicity Test Program for Pseudo-Random Number Generator Routines on the HP-41C

Charles T. Tart
Univeristy of California, Davis

The availability of high quality, relatively inexpensive programmable scientific calculators, such as the Hewlett-Packard HP-41C, offers a valuable new tool to parapsychologists. As described in another paper (Tart, 1982), the HP-41C can be programmed to provide a laboratory quality ESP test and feedback training instrument. A rapid evolution of features available can be expected in such "calculators" (they are really hand-held computers), so they may play a strong role in parapsychological research in the near future.

In using these instruments for ESP testing devices, it is essential that they have high quality subroutines for generating random numbers. At present we are limited to pseudo-random number generator (PRNG) subroutines, but with adequate algorithms for such routines and frequent changing of quasi-random seed values, it should not be too difficult to generate adequately random target numbers. This paper describes a test routine for evaluating a PRNG. The specific program is for the HP-41C, but it should be readily adaptable to any programmable scientific calculator.

A true RNG has two important properties. First, the probability of any output is equal to that of any other output, so as a series gets longer the proportion of any particular output to the total output approaches 1/C, where C is the number of output choices of the RNG. Second, there is no sequential dependency between numbers, that is, the probability of any number following some previous output number or sequence of such numbers is equal to that of any other number following that previous output number or sequence of numbers. This also means that no matter what output numbers have already been generated, they give you no useful way of predicting what the next outputs will be. (For fuller discussion of the non-predictability criterion see Tart, 1979a; 1979b; Tart & Dronek, 1980).

The adequacy of a RNG or PRNG can be evaluated statistically by testing a large sample of outputs for equiprobability and lack of sequential dependency. A common way of doing this testing is to count up not only single outputs (Ø, 1s, 2s, etc., called singlets), but also sequential outputs (Ø followed by Ø, Ø followed by 1, Ø followed by 2, etc., called doublets). Sequential output testing usually includes doublets, and, if there is a theoretical reason to suspect higher order sequential dependencies, triplets, quadruplets, etc.

The following test program was designed to check the PRNG subroutine used in an ESP test program described elsewhere (Tart & Puthoff, 1981; Tart, 1982), where nonpredictability is extremely important. It requires a printer and extended memory. By putting in your own PRNG subroutine as step Ø56 and calling it LBL Ø5, substituting for steps Ø56 through Ø66 in the present program, you can test it.

The test program assumes that your PRNG produces an integer (no fractional part) output, which appears as step 67 in this test program. The particular PRNG subroutine written into this program starts with a seed number that is stored in register Ø9. The algorithm is described in Tart, 1982. It produces a fractional output between .99999999 and .ØØØØØØØ1, but the scaling factor (number of choices) stored in register Ø8 scales this up to the proper choice range, and the INT (integer) function in the HP-41C discards the fractional part of the number produced. I use the time, to the nearest second, as a quasi-random way of getting a seed number for each run of the PRNG that is relatively independent of any deliberate control on my part.

Operation:

Detailed operation notes are listed with the program. Size Ø61 should be executed before running it. Briefly, the TESTRNG program starts with housekeeping chores of clearing registers and flags, then prompts for the total number of outputs, trials, wanted from the PRNG (TOTAL N?) in this run, for the number of choices (2 to 1Ø) the PRNG is to have, whether you want the raw PRNG output printed (enter "N" for no, otherwise just press R/S), and whether you want the PRNG output accumulated for later statistical analysis (ANALYZE?) (again enter "N" for no, otherwise R/S).

A note on limitations. A singlet analysis can be carried out if your PRNG output is Ø to 9, but TESTRNG can carry out a doublet analysis only if there are no more than 5 choices (outputs Ø to 4). Doublet analysis will be automatically skipped if C > 5. This is due to the difficulty of addressing enough registers: as it is, registers ØØ through 6Ø (size Ø61) are used. In general I suspect that if your PRNG rountine shows no singlet or doublet biases for outputs Ø through 4 it probably doesn't have them for outputs 5 through 9, but don't count on it.

The TESTRNG program then prompts for a time or other seed number (SEED?), following which it accesses the PRNG subroutine until it has collected N outputs. This can take a while. For my particular PRNG subroutine, it takes about 4 minutes to collect 1ØØ outputs. When N outputs are present the total number of outputs to date are printed ($\Sigma\Sigma=$) and a beep signals that a new seed is wanted. Total outputs equal N on the first run of TESTRNG. This feature exists because I usually use my PRNG in the ESP test program for runs of 25 trials or less and then enter a new time seed for each run to assure better randomicity. Your N should be the usual number of outputs you use at a time from your PRNG, or an even larger sample.

At this point you can enter a new time seed and get N more outputs from the PRNG. For analysis purposes, these outputs are accumulated with the preceeding batches of N outputs. For example, if you used a binary PRNG and in 1ØØ trials had 48 Øs and 52 1s, then ran another 1ØØ trials with 46 Øs and 54 1s, the storage registers would cumulate 94 Øs and 1Ø6 1s.

If you elected to store data for analysis, at the end of any run of N trials you can XEQ "CHI" to start analysis of the cumulated results. This analysis does not affect data storage registers, so you can cumulate more data after such an analysis. CHI is a Chi-square statistical test at the singlet and (if $C \leq 5$) the doublet levels. The resulting values of Chi-square can be looked up in any table. Note also that if the expected value of any particular PRNG output is less than 5, the Chi-square test is generally not valid: the TESTRNG program will automatically print "E < 5" to alert you when this happens.

Example:

Here's an example of TESTRNG's operation. Run 2Ø trials of a 3-choice PRNG and analyze, using the current PRNG subroutine. The time seed is 935.54. The PRNG raw output printed is

```
2. 1. 2. Ø. 1. 2.
1. 1. 1. Ø. 1. Ø.
1. 2. 1. 2. Ø. 2.
2. 1.
```

Upon executing "CHI" the printer gives us

```
4. 9. 7.
S CHI SQ = 1.9ØØ
```

indicating that Ø was generated 4 times, 1 was generated 9 times, and 2 was generated 7 times. The Chi-square analysis uses the formula

$$\text{Chi-square} = \sum \frac{(O-E)^2}{E}$$

For each possible output category (Ø, 1, or 2 in this case) the observed number of appearances of that output (O) has the expected number of appearances (E, 2Ø/3 in this case) subtracted from it, the result is squared and then divided by E, and the results are summated for all possible outputs.

The doublet analysis subroutine then prints out

```
  Ø. 3. 1.
  2. 2. 4.
  2. 4. 1.
D CHI SQ = 7.Ø53
  E < 5
```

indicating that an output of Ø was followed by Ø zero times, Ø was followed by 1 three times, 1 was followed by Ø two times, etc. Chi-square is computed by the same formula, but note that we only have 19 doublets in 2Ø trials, and we sum over 9 output categories. Because the expected frequencies in each category were less than 5, "E < 5" is printed out as a warning that this particular analysis is not valid.

Testing External Generators:

The TESTRNG program can also be used to analyze an externally generated set of random numbers with the following modification. For the current steps 58 to 67, put in instead

```
58 "NUMBER?"
59 PROMPT
```

The program will renumber itself in the HP-41C so the next step 6Ø will be the former step 68, viz. STO 16. On each cycle, which takes about a second, the program will prompt you to enter an integer number, "NUMBER?" Enter your external generator's output and press R/S.

I hope other researchers will find this program useful.

The TESTRNG Program:

Ø1 LBL "TESTRNG"	
Ø2 CF ØØ	Clear flags, store loop control #
Ø3 CF Ø1	for register clear.
Ø4 CF Ø2	
Ø5 CF Ø3	
Ø6 Ø.Ø59Ø1	
Ø7 STO 6Ø	
Ø8 LBL Ø2	Loop for storing Øs in register
Ø9 Ø	ØØ through 59.
1Ø STO IND 6Ø	
11 ISG 6Ø	
12 GTO Ø2	
13 FIX Ø	
14 "TOTAL N?"	
15 PROMPT	
16 STO Ø5	
17 LBL Ø7	Prompts for number of outputs of
18 "CHOICES?"	PRNG, choices, C.
19 PROMPT	
2Ø STO Ø8	
21 1	Computes and stores C-1 for later
22 -	computational ease.
23 STO 46	
24 1Ø	Limits C to maximum of 1Ø.
25 RCL Ø8	
26 X>Y?	
27 GTO Ø7	
28 5	Test: C > 5? If so, no doublet analysis data
29 RCL Ø8	will be stored.
3Ø X>Y?	
31 SF Ø2	
32 "N"	Should raw PRNG ouput be printed? Enter "N"
33 ASTO Y	if not, R/S if you want it.
34 AON	
35 "PRINT RAW?"	
36 PROMPT	
37 AOFF	
38 ASTO X	
39 X=Y?	
4Ø SF ØØ	

41 "N"	Should singlet and doublet data be stored for analysis? Enter "N" if not, R/S if you want it.
42 ASTO Y	
43 AON	
44 "ANALYZE?"	
45 PROMPT	
46 AOFF	
47 ASTO X	
48 X=Y?	
49 SF Ø1	
5Ø LBL Ø1	Prompts for a seed value for the PRNG.
51 TONE 6	
52 "SEED?"	
53 PROMPT	
54 LN	
55 ABS	
56 STO Ø9	
57 LBL Ø5	Pseudo-Random Number Generator, PRNG routine. Your routine should be entered here. This one takes transformed seed from reg. Ø9, adds pi, raises sum to 5th power and stores fractional part in reg. Ø9. Lines 65-67 scale result to range of C and take integer.
58 PI	
59 RCL Ø9	
6Ø +	
61 5	
62 Y↑X	
63 FRC	
64 STO Ø9	
65 RCL Ø8	
66 *	
67 INT	
68 STO 16	
69 FS? ØØ	Test: accumulate PRNG output for later analysis?
7Ø GTO Ø4	
71 LBL Ø3	Accumulate PRNG output in print buffer, with spaces, for later printing.
72 ACX	
73 1	
74 SKPCHR	
75 LBL Ø4	Increment trials counter (reg. Ø6) by one.
76 1	
77 ST+ Ø6	
78 FS? Ø1	Test: analysis wanted?
79 GTO Ø6	
8Ø RCL 16	Add 5Ø to PRNG output number to get control number so proper singlet count register will be incremented. Increment by one.
81 5Ø	
82 +	
83 STO 17	
84 1	
85 ST+ IND 17	
86 FS? Ø2	Test: doublet analysis OK?
87 GTO 11	

88 RCL Ø6	Test: first trial of run? If so, skip
89 1	lines 91-96 to increment a doublet
9Ø X=Y?	register.
91 GTO Ø6	
92 RCL 18	Add 1Øx previous PRNG output to current
93 RCL 16	PRNG ouput to determine doublet storage
94 +	register number.
95 STO 19	
96 1	Increment appropriate doublet register
97 ST+ IND 19	by one.
98 LBL Ø6	Multiply current PRNG output by 1Ø, store,
99 RCL 16	use for doublet increment addressing on
1ØØ 1Ø	next trial.
1Ø1 *	
1Ø2 STO 18	
1Ø3 LBL 11	Test: end of run? If not, activate another
1Ø4 RCL Ø6	PRNG output.
1Ø5 RCL Ø5	
1Ø6 X>Y?	
1Ø7 GTO Ø5	
1Ø8 PRBUF	Print accumulated PRNG output. Add trials
1Ø9 RCL Ø6	of current run to grand trials counter.
11Ø ST+ Ø7	
111 RCL Ø7	Print grand total of PRNG trials to date.
112 "ΣΣ = "	
113 ARCL X	
114 AVIEW	
115 Ø	Reset trials counter to Ø.
116 STO Ø6	
117 GTO Ø1	
118 LBL "CHI"	CHI-SQUARE ANALYSES
119 Ø	
120 STO 26	Clear Chi-square total registers.
121 STO 35	
122 5Ø	Compute control number for indirect RCL of
123 STO 27	singlet registers 5Ø through 5Ø+(C-1).
124 RCL 46	
125 +	
126 1 E3	# = (5Ø+ [C-1]) / 1,ØØØ + .ØØØØ1
127 /	
128 1 E-5	
129 +	= 5Ø.ØØ(C-1)Ø1
13Ø ST+ 27	
131 STO 28	
132 RCL Ø7	Computed expected singlet frequency, E,
133 RCL Ø8	where
134 /	E=N/C
135 STO 25	

136 5 137 X>Y? 138 SF Ø3	Test: E<5? If so, SF Ø3.
139 LBL Ø8 14Ø RCL IND 27 141 ACX 142 1 143 SKPCHR	Accumulate frequencies of various singlets for later printing.
144 RCL IND 27 145 RCL 25 146 - 147 X↑2 148 RCL 25 149 / 15Ø ST+ 26 151 ISG 27 152 GTO Ø8	Compute singlet Chi-square $\text{Chi-square} = \sum \frac{(O-E)^2}{E}$ O = observed frequency, each output E = expected frequency of each PRNG output
153 FIX 3 154 PRBUF 155 RCL 26 156 "S CHI SQ= " 157 ARCL X 158 AVIEW 159 PSE 16Ø CLA 161 FIX Ø	Print singlet Chi-square.
162 FS? Ø3 163 XEQ 13	Test: E<5? If so, print "E<5".
164 CF Ø3 165 FS? Ø2 166 GTO 12	Test: doublet analysis OK?
167 RCL Ø7 168 ENTER↑ 169 ENTER↑ 17Ø RCL Ø5 171 / 172 -	Compute reduced N for doublet analysis (one trial lost on each run).
173 RCL Ø8 174 X↑2 175 / 176 STO 25	Compute expected doublet frequency, E.
177 5 178 X>Y? 179 SF Ø3	Test: E<5? If so, SF Ø3.

18Ø RCL 46	Compute control number for indirect RCL
181 1 E3	of doublet registers ØØ through Ø(C-1),
182 /	1Ø through 1(C-1), etc.
183 1 E-5	
184 +	
185 STO 27	
186 STO 29	
187 LBL Ø9	Nested loops for computing Chi-square
188 LBL 1Ø	values and printing doublet frequency
189 FIX Ø	table. LBL 1Ø moves across a row of
19Ø RCL IND 27	the table, LBL Ø9 moves down to
191 ACX	the next row. Lines 2Ø4 to 2Ø9 change
192 1	the loop control value to step to the next
193 SKPCHR	row.
194 RCL IND 27	
195 RCL 25	
196 -	
197 X↑2	
198 RCL 25	
199 /	
2ØØ ST+ 35	
2Ø1 ISG 27	
2Ø2 GTO 1Ø	
2Ø3 PRBUF	
2Ø4 1Ø	
2Ø5 RCL Ø8	
2Ø6 -	
2Ø7 .Ø1	
2Ø8 +	
2Ø9 ST+ 27	
21Ø ISG 29	
211 GTO Ø9	
212 FIX 3	Print doublet Chi-square
213 RCL 35	
214 "D CHI SQ= "	
215 ARCL X	
216 AVIEW	
217 PSE	
218 CLA	
219 FIX Ø	
22Ø FS? Ø3	Test: E<5? If so, print "E<5".
221 XEQ 13	
222 CF Ø3	Go to seed prompt routine to generate
223 GTO Ø1	another run.
224 LBL 12	Print "No DBLT ANAL" if C>5.
225 "NO DBLT ANAL"	
226 AVIEW	
227 PSE	
228 GTO Ø1	

```
229 LBL 13        Print "E<5" to warn that Chi-square analysis
230 SF 12           probably not valid.
231 "E<5"
232 AVIEW
233 CF 12
234 RTN
235 END
```

References:

Tart, C., Randomicity, predictability, and mathematical inference strategies in ESP feedback experiments: Discussion of Dr. Gatlin's paper. J. Amer. Soc. Psych. Res., 1979, 73, 44-60.

Tart, C., Could mathematical inference strategies have inflated scoring in the first feedback training study? In W. Roll (Ed.), Research in Parapsychology 1978. Metuchen, N.J.: Scarecrow Press, 1979. Pp. 128-129.

Tart, C., Paper submitted for 1982 Parapsychological Association meeting.

Tart, C., & Dronek, E., Trying to profit from non-randomicity in ESP target sequences: Initial explorations with the Probablistic Predictor Program. In W. Roll (Ed.), Research in Parapsychology 1979. Metuchen, N.J.: Scarecrow Press, 1980. Pp. 86-88.

Tart, C., & Puthoff, H., ESP testing and training. PPC Calculator Journal, 1981, 8, No. 6, 62-63.