

Seed That Apple RND Generator!

by Darnell Salyer*

Do your programs use the random number generator of the Apple? Have you "seeded" the generator to be sure the same series of random numbers is not used during each execution on your freshly activated microcomputer?

Writers of many computer assisted instructional programs for chemistry, including commercially available programs, have not taken measures to see that random number selection begins at a randomly chosen place in the fixed sequence of numbers accessed by RND(n). The oversight is quite likely if a program written for some other computer is modified for use on the Apple.

If an instructional program uses RND a variable number of times, as for example in branched tutorials which depend on user's answers or responses, a repetition of the sequence of random numbers used in the program will not be so readily apparent except when all responses are correct or all are incorrect.

If a CAI program calls for random numbers a fixed number of times regardless of the number of correct or incorrect responses, and if RND is not seeded, then a student user would see the same examples, unknown number, molecular weights or other data values, comments, "readings" in instrument simulations, graphics features, etc. each time he/she sits down at the microcomputer, turns it on, and runs the program. In like manner, two or more users would see identical CAI items upon first executions of a program.

Although dialects of Basic for minicomputers such as the Digital PDP 11/70 and for some micros, e.g. Radio Shack TRS-80, typically provide that a statement such as RANDOM or RANDOMIZE will achieve the seeding for an accompanying RND statement, that same provision was not made for the Apple. Unfortunately, this is not made clear in most of the Basic reference manuals this writer has seen. The Atari produces nonrepetitive sequences with RND (Ø) while RND(-TI) for Commodore machines takes its seed from an internal timer.¹ The assertion that "every time" RND is used with any positive argument a new random number from Ø to 1 is generated is incorrect at least with respect to initial start-up. It should also be understood that simply changing the positive values for dummy argument n of RND(n) makes no difference since all give the same sequence upon initial use.

In an appendix of one widely used manual for the Apple,² the intrinsic subroutine executed by CALL -756 is described very briefly. This will place a number between 1 and 255 in specified memory locations 78 or 79. The call executes as an input and the "seed" is placed when the program user strikes a key. Since most programs will usually contain early input statements, such as those asking for "press return" after directions, the call may easily be added to serve as the input.

```
1110 PRINT "PRESS RETURN WHEN READY TO CONTINUE";
1120 CALL -756
1130 FOR XX = 1 TO PEEK (78): TA = RND(5): NEXT XX
```

For the Apple II+ a seed may be obtained from (78) or (79) even without the call, as long as the program has had one or more inputs prior to peeking. In such cases line 1130 alone gives entry to the random number sequence at one of 255 different places. A seed number of values is essentially thrown away and the next value would be the first used by the main program. The seed may be adjusted arithmetically.

In addition to CALL -756 another short machine language program was recently described which seeds the Apple RND and requires a keystroke.³

Other means of seeding:

- a) If a program has required entry of a student name, the date, time, numbers or other information, the seed may be taken from an appropriate ASCII or VAL statement.
- b) If game paddles are in use a PDL(n) value may be read and used.
- c) A separate small auxiliary file containing a seed could be used with the program. The program opens the file, reads the number, increments the number, writes it back into the file and closes. Also, such a file could provide a record of program usage.

The seed obtained by any of these procedures is then used in some manner, as in line 1130 above, before the main program uses RND(n) for the first time. Diskettes with a menu for program selec-

tion such as those of the Seraphim project may have the generator seeding provision places in the menu to facilitate any program selected.

Sometimes it may not be possible to add the call and/or needed lines, especially if a program is copy protected or unlistable. It may be possible to execute seeding statements in immediate mode or with a short program from the same or another diskette, and then, without turning the computer off, RUNning or BRUNning the main program or the menu.

*Chemistry Department
Eastern Kentucky University
Richmond, KY 40475

1. Mansfield, Richard (ed.), "Zones of Unpredictability", in COMPUTE!, 5(11), 146-47 (1983).
2. Poole, Lon et al., Apple II User's Guide, Appendix D, Osborne/McGraw Hill, 1981.
3. Isaacson, Mark, "The Missing Applesoft Random Number", in The Apple Orchard, 4(6), 76-77 (1983).

SERAPHIM-CCCE HIGH SCHOOL TEACHER WORKSHOP SERIES

The workshops have two basic objectives:

- To illustrate how microcomputers can be of use to a chemistry teacher
- To provide an opportunity to preview representative samples of commercially available software.

This is accomplished with a collection of software and supporting materials which currently includes all the SERAPHIM disks plus 150 commercial programs from 15 different publishers.

The guidelines for organizing a workshop are as follows:

PROJECT SERAPHIM and the CCCE will provide:

1. A set of software and supporting printed materials.
2. Publicity in national publications.
3. A qualified workshop leader, when none is locally available.

The host institution is responsible for providing:

1. A minimum of 10 Apple computers for every 20 participants.
2. One large screen (23-25") color TV/monitor for every 15 participants with proper connectors to mate with the Apple output.
3. An overhead projector and screen.
4. Adequate meeting rooms to accomodate both the demonstration and the hands-on parts of the workshop.
5. An assistant who is familiar with the hardware and facilities.
6. Controls to make sure that copyrighted materials are NOT COPIED.

The cost of the workshop is \$10 per person with a minimum charge of \$200. If necessary, the workshop organizer can assess an additional fee to cover local costs (for refreshments, lunch, etc.). The CCCE will pay all shipping costs and expenses incurred in providing a workshop leader. For further information and application forms, contact Paul Cauchon at Canterbury School, New Milford, CT 06776, (203) 355-3452.

Hardware QUERIES

Send Hardware queries, rebuttals and information to Ken Ratzlaff, Instrumentation Design Laboratory, Chemistry Department, University of Kansas, Lawrence, KS 60045 (413) 864-3754.

Software QUERIES and REPLIES

Software queries and replies should be sent to Ken Loach, Department of Chemistry, SUNY College, Plattsburgh, NY 12901, (818) 564-2230.

For a long time, chemists have had to make do with general purpose computer languages and software, written without the chemist's needs in mind. A pleasing feature of mid-80's computing is the growing number of lab-related high-level software systems now coming on to the market.