An extract from Donald E. Knuth's book *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.*

It is not easy to invent a foolproof source of random numbers. This fact was convincingly impressed upon the author in 1959, when he attempted to create a fantastically good generator using the following peculiar approach:

Algorithm K (Super-random number generator). Given a 10-digit decimal number X, this algorithm may be used to change X to the number that should come next in a supposedly random sequence. Although the algorithm might be expected to yield quite a random sequence, reasons given below show that it is not, in fact, very good at all. (The reader need not study this algorithm in great detail except to observe how complicated it is; note, in particular, steps K1 and K2.)

K1. [Choose number of iterations.] Set $Y = [X/10]$, the most significant digit of X. (We will execute steps K2 through K13 exactly $Y + 1$ times; that is, we will apply randomizing transformations a random number of times.)

K2. [Choose random step.] Set $Z = [X/10] mod 10$, the second most significant digit of X. Go to step K(3 + Z). (That is, we now jump to a random step in the program.)

K3. [Ensure $X \geq 5 \times 10^9$.] If $X < 5000000000$, set $X = X + 5000000000$.

K4. [Middle square.] Replace X by $X^2/10^5 mod 10^{10}$, that is, by the middle of the square of X.

K5. [Multiply.] Replace X by $(1001001001X) mod 10^{10}$.

K6. [Pseudo-complement.] If $X < 100000000$, then set $X = X + 9814055677$; otherwise set $X = 10^{10} - X$.

K7. [Interchange halves.] Interchange the low-order five digits of X with the high-order five digits; that is, set $X = 105(X mod 10^5) + [X/10^5]$, the middle 10 digits of $(10^{10} + 1)X$.

K8. [Multiply.] Same as step K5.

K9. [Decrease digits.] Decrease each nonzero digit of the decimal representation of X by one.

K10. [99999 modify.] If $X < 10^5$, set $X = X^2 + 99999$; otherwise set $X = X - 99999$.

K11. [Normalize.] (At this point X cannot be zero.) If $X < 10^9$, set $X = 10X$ and repeat this step.

K12. [Modified middle square.] Replace X by $[X(X - 1)/10^5 mod 10^{10}$, that is, by the middle 10 digits of $X(X - 1)$.

K13. [Repeat?] If $Y > 0$, decrease Y by 1 and return to step K2. If $Y = 0$, the algorithm terminates with X as the desired 'random' value.

(The machine-language program corresponding to this algorithm was intended to be so complicated that a person reading a listing of it without explanatory comments wouldn't know what the program was doing.)

Considering all the contortions of Algorithm K, doesn't it seem plausible that it should produce almost an infinite supply of unbelievably random numbers? No! In fact, when this algorithm was first put onto a computer, it almost immediately converged to the 10-digit value 6065038420, which, by extraordinary coincidence, is transformed into itself by the algorithm (see Table 1 in https://www.informit.com/articles/article.aspx?p=2221790). With another starting number, the sequence began to repeat after 7401 values, in a cyclic period of length 3178.