[6.55] Algorithms for factor() and isPrime()

The following description of the algorithms for *factor()* and *isPrime()* was obtained from TI by Bhuvanesh Bhatt. It is included with TI's permission.

"The algorithms used in the TI-92, TI-92 Plus, and TI-89 are described in D. Knuth, The Art of Computer Programming, Vol 2, Addison-Wesley, but here is a brief description:

"The TI-92 simply divides by successive primes through the largest one less than 2^16.1: It doesn't actually keep a table or use a sieve to create these divisors, but cyclically adds the sequence of increments 2, 2, 4, 2, 4, 2, 4, 6, 2, 6 to generate these primes plus a few extra harmless composites.

"TI-92 Plus and TI-89 start the same way, except that they stop this trial division after trial divisor 1021, then switch to a relatively fast Monte-Carlo test that determines whether the number is certainly composite or is almost certainly prime. (Even knowing the algorithm, I cannot construct a composite number that it would identify as almost certainly prime, and I believe that even centuries of continuous experiments with random inputs wouldn't produce such a number).

"The *isPrime()* function stops at this point returning either false or (almost certainly) true.

"If the number is identified as composite, the *factor()* function then proceeds to the Pollard Rho factorization algorithm. It is arguably the fastest algorithm for when the second largest prime factor has less than about 10 digits. There are faster algorithms for when this factor has more digits, but all known algorithms take expected time that grows exponentially with the number of digits. For example, the Pollard Rho algorithm would probably take centuries on any current computer to factor the product of two 50-digit primes. In contrast, the *isPrime()* function would quickly identify the number as composite.

"Both the compositivity test and the Pollard Rho algorithm need to square numbers as large as their inputs, so they quit with a *"Warning: ... Simplification might be incomplete."* if their inputs exceed about 307 digits."