RANDOM.TEX ver 0.2 (Donald Arseneau)

Generating "random" numbers in TEX.

`\setrannum{`⟨*counter*⟩`}{`⟨*minimum*⟩`}{`⟨*maximum*⟩`}`
`\setrandimen{`⟨*dimen-register*⟩`}{`⟨*minimum*⟩`}{`⟨*maximum*⟩`}`

Random integers are generated in the range 1 to 2147483646 by the macro `\nextrandom`. The result is returned in the counter `\randomi`. Do not change `\randomi` except, perhaps, to initialize it at some random (or specific) value. If you do not initialize it, it will be initialized using the time and date. (This is a sparse initialization, giving fewer than a million different starting values, but you should use other sources of numbers if they are available—just remember that most of the numbers available to TEX are not at all random.)

The `\nextrandom` command is not very useful by itself, unless you have exactly 2147483646 things to choose from. Much more useful is the `\setrannum` command which sets a given counter to a random value within a specified range. There are three parameters:

`\setrannum{`⟨*counter*⟩`}{`⟨*minimum*⟩`}{`⟨*maximum*⟩`}`

For example, to simulate a die-roll:

```
\setrannum{\die}{1}{6} \ifcase\die...
```

If you need random numbers that are not integers, you will have to use dimen registers instead with `\setrandimen`. For example, to set a random page width between 3 and 6.5 inches:

```
\setrandimen\hsize{3in}{6.5in}
```

The "`\pointless`" macro will remove the "`pt`" that TEX gives so you can use the dimensions as pure "real" numbers. In that case, specify the range in `pt` units. For example,

```
\setrandimen\answer{2.71828pt}{3.14159pt}
The answer is \pointless\answer.
```

The random number generator is the one by Lewis, Goodman, and Miller (1969) and used as "ran0" in "Numerical Recipies" using Schrage's method for avoiding overflows. The multiplier is 16807 (7^5), the added constant is 0, and the modulus is 2147483647 ($2^{31} - 1$).

See CACM, Vol. 36, no. 7, (July 1993), p. 109. The original authors Park and Miller have since concluded that a better multiplier is 48271, rather than their original 16807.

The range of integers generated is $1 - 2147483646$. A smaller range would reduce the complexity of the macros a bit, but not much—most of the code deals with initialization and type-conversion. On the other hand, the large range may be wasted due to the sparse seed initialization.