Pascal: Real number format

Real numbers in UCSD Pascal are floating point numbers between +/-1.17550E-38 and +/-3.40282E+38, inclusive. Real numbers take up four bytes (2 words) of storage. Their binary representation is similar to the proposed IEEE standard for floating point numbers:

```
  31   30 . . . . . 23  22 . . . . . . . . . 0  <== 32 bits
  Sign      Exponent                Mantissa
```

"Mantissa" is the name given to the decimal portion of a number; by convention, it's expressed in scientific (exponential) notation. The "exponent" indicates the power to which the mantissa is raised; it's represented in base 2 (2^N). The number 3×10^2, for example, is defined as having a mantissa of 3 and an exponent of 2, in base 10 (decimal).

The sign bit refers to the sign of the mantissa; it's 0 if positive, 1 if negative. The exponent is "offset" by 127--that is, a value of 127 in the exponent field corresponds to an exponent of 0. Similarly, if the value is 1, the exponent is -126, and if the field is 254, the exponent is +127. A value of 0 indicates that the real number is 0.

The mantissa of the real number is stored in normalized format in bits 0-22. "Normalizing" a number means adjusting it so that the highest bit is significant (that is, set to 1). The exponent indicates how many times, and in which direction, the value was shifted during normalization.

Notice that the MSB of the mantissa of any non-zero number that has been normalized is always a one. Zero can be treated as a special case--the exponent is simply set to zero. For the sake of additional precision, then, the mantissa has an implied "1" that is not stored, resulting in a functional 24-bit mantissa, even though only 23 bits are actually used. This structure yields slightly more than a 6-decimal-place (single precision) accuracy.

Real numbers may be formatted for output by means of field-width designations. As described on pp. 36-37 of the Apple Pascal language Reference Manual, the output specification has the following form:

        Real : FieldWidth : FractionLength

where FieldWidth is the minimum number of characters written, including the decimal point (default=1). FractionLength is the number of digits to be written after the decimal place (default=5). Thus, a field specification of R:8:3 indicates the real variable R, printed within a field size of 8, with 3

of those digits appearing to the right of the decimal. A FractionLength of zero is illegal.

If the field size necessary for displaying the variable accurately is greater than the formatting specification, the formatting is ignored. If the size is smaller than FieldWidth, the field is padded with blanks to the left of the variable; the variable is thereby right-justified.
Apple Tech Notes

Keywords: <None>

==

This information is from the Apple Technical Information Library.

19960215 11:05:19.00

Tech Info Library Article Number: 691