Addition to the APL 90 for Macintosh reference manual.

Floating point arithmetic

APL 90 uses native floating-point arithmetic provided by the SANE package, and implemented either in the ROM of the Macintosh plus, or in the System file of the old Macs. The SANE package follows the ANSI/IEEE standard for Binary Floating-Point Arithmetic (Std 754-1985). As required by the satndard, this package handles numbers, two infinities (∞ and –∞), and NaNs ("not a number").

The following options have been chosen in APL 90 regarding infinities and NaNs.

Printing
When formating or printing Infinities and NaNs, these objects appear as the character strings "Inf", "¯Inf", and "NaN(*n*)", where *n* is a number between 0 and 255.

Comparisons
Equal and Different. These functions apply to numbers, ∞s and NaNs. Numbers are compared using comparison tolerence. Other objects are equal only if they are the same object (same Inf, or the same NaN).
Other comparison functions. The other functions apply only to numbers and Infs. A domain error is signaled if one of the operands is a NaN.

Maximum and minimum
Maximum and Minimum work as expected on numbers. ∞ is the result of Maximum when one of the operands is ∞. –∞ is the result of Minimum when one of the operands is –∞. When one operand is a NaN, it becomes the result.

Sign
The signs of ∞ and –∞ are 1 and ¯1 respectively. The sign of NaNs is defined to be 0.

Absolute value
The absolute value of ∞ and –∞ is ∞. The absolute value of a NaN is the NaN itself.

New functions
Three new functions are provided to deal with NaNs and ∞.
The ≠ function, used in its monadic form, returns 1 if its operand is a NaN, 0 otherwise. This function apply only to numbers.
The INF function (MLDF), applied to any number, returns ∞ with the sign of this number.
The NAN function (MLDF), applied to any integer between 0 and 255 returns the corresponding NaN. For example, the result of NAN 25 will print as "NaN(25)".

Brace notation

Brace notation is fully implemented in APL 90. Brace notation allows creation of vectorial constants of any type.
The following lines contain pairs of equivalent expressions, although the right expression is a notation, and involves no computation at execution time, while the left expression may necessitate the application of stand notation, or the execution of one or more APL function. Note also that parentheses are not needed around constants expressed with brace notation, while they may be needed around the equivalent expression.

,3	{3}
2 3 6	{2 3 6}
,K□K	{K□K}
K□bcK	{K□K KbK KcK}
2 4 KxK KzK	{2 4 KxK KzK}
:,2" K□zK	{{2} K□zK}
:1 2 3" :4 5"	{{1 2 3}{4 5}}
ZZZZ,KxK	{{{{{KxK}}}}}
KK :I0"	{KK Ë}

The following constants cannot be expressed without brace notation :

{} An empty vector of type atom. The “first” of this vector is the empty atom, or *nilde*, that prints as °°
{hello world} A vector of two atoms.
{□[5-n} A five element vector : the atom □, the atom [, the number 5, the atom - and the atom n.