APL is a terrific language for manipulating numeric and textual information, with an unrivalled range of **arithmetic** and data processing primitive functions and operators. However, when it comes to **algebraic** manipulation, APL is pretty useless. APL seems quite sufficient for business information, so who needs algebra? Not accountants (yet), but scientists and engineers need algebra. They perform algebraic calculations in REDUCE that would take centuries to complete by hand. One might propose that APL and REDUCE learn from one another. They are not very dissimilar in form. Both are array-oriented, and can be united on paper by doing away with VALUE ERROR and returning an expression rather than a value.

APL	REDUCE
a←1 2 3 a 1 2 3	a:=mat((1,2,3)); a; [1 2 3]
a+.×⍉a 14	a*tp a; [14]
*1⋄⍟*1⋄○1⋄2○1 2.718281828 1 3.141592654 ¯1	e;ln e;pi;cos pi; 2.71828182846 1 3.14159265359 -1
p←1 :For i :In ⍳50 ⋄ p×←1 ⋄ :EndFor p	for i:=1:50 product i 30414093201713378043612608166O6...
!50 3.04140932017E64	factorial 50 30414093201713378043612608166O6...
but.. (x+y)*3 VALUE ERROR	(x+y)^3; 3 2 2 3 x + 3*x *y + 3*x*y + y