= 2100 =
1 @,@ 267 @,@ 650 @,@ 600 @,@ 228 @,@ 229 @,@ 401 @,@ 496 @,@ 703 @,@ 205 @,@ 376
This series of steps only requires 8 multiplication operations instead of 99 (since the last product above takes 2 multiplications) .
In general , the number of multiplication operations required to compute bn can be reduced to ? (log n) by using exponentiation by squaring or (more generally) addition @-@ chain exponentiation . Finding the minimal sequence of multiplications (the minimal @-@ length addition chain for the exponent) for bn is a difficult problem for which no efficient algorithms are currently known (see Subset sum problem) , but many reasonably efficient heuristic algorithms are available .

= = Exponential notation for function names = =

Placing an integer superscript after the name or symbol of a function , as if the function were being raised to a power , commonly refers to repeated function composition rather than repeated multiplication . Thus , f 3 (x) may mean f (f (f (x))) ; in particular , f ? 1 (x) usually denotes the inverse function of f . Iterated functions are of interest in the study of fractals and dynamical systems . Babbage was the first to study the problem of finding a functional square root f 1 / 2 (x) .
For historical reasons , this notation applied to the trigonometric and hyperbolic functions has a specific and diverse interpretation : a positive exponent applied to the function 's abbreviation means that the result is raised to that power , while an exponent of ? 1 denotes the inverse function . That is , sin2 x is just a shorthand way to write (sin x) 2 without using parentheses , whereas sin ? 1 x refers to the inverse function of the sine , also called arcsin x . Each trigonometric and hyperbolic has its own name and abbreviation both for the reciprocal ; for example , 1 / (sin x)
= (sin x) ? 1 =
csc x , as well as for its inverse , for example cosh ? 1 x = arcosh x . A similar convention applies to logarithms , where log2 x usually means (log x) 2 , not log log x .

= = In programming languages = =

The superscript notation xy is convenient in handwriting but inconvenient for typewriters and computer terminals that align the baselines of all characters on each line . Many programming languages have alternate ways of expressing exponentiation that do not use superscripts :
x ? y : Algol , Commodore BASIC
x ^ y : BASIC , J , MATLAB , R , Microsoft Excel , Analytica , TeX (and its derivatives) , TI @-@ BASIC , bc (for integer exponents) , Haskell (for nonnegative integer exponents) , Lua and most computer algebra systems
x ^ ^ y : Haskell (for fractional base , integer exponents) , D
x * * y : Ada , Bash , COBOL , CoffeeScript , Fortran , FoxPro , Gnuplot , OCaml , F # , Perl , PHP , PL / I , Python , Rexx , Ruby , SAS , Seed7 , Tcl , ABAP , Mercury , Haskell (for floating @-@ point exponents) , Turing , VHDL
pown x y : F # (for integer base , integer exponent)
x ? y : APL
Many programming languages lack syntactic support for exponentiation , but provide library functions .
In Bash , C , C + + , C # , D , Go , Java , JavaScript , Perl , PHP , Python and Ruby , the symbol ^ represents bitwise XOR . In Pascal , it represents indirection . In OCaml and Standard ML , it represents string concatenation .

= = List of whole @-@ number powers = =