# Beyond The Four Fours

an investigation by Hugh Darwen and Gerard Joseph

In recreational mathematics the "four fours" problem invites the solver to express every natural number from 1 to 100 using just four 4s and the operators +, −, ×, ÷, √, ! (factorial), and exponentiation. Also admitted are the decimal point (·) and dot-above (˙) to indicate a recurring decimal, and parentheses to specify the order of operations.

We investigated similar problems for every combination of four nonzero digits (not necessarily distinct), 495 in all.

For added interest, we tried to find what we called **clean** solutions wherever possible. A clean solution is one that avoids redundant expressions of the following kinds:

1. Multiplication or division of a quantity by 1 ($x \times 1$, $x \div 1$)
2. Addition/subtraction of zero to/from a quantity ($x \pm 0$)
3. Explicit expression of a quantity as the 1st power ($x^1$) or 1st root ($\sqrt[1]{x}$) of itself
4. Use of 1 as an explicit power or root of itself ($1^x$ or $\sqrt[x]{1}$)
5. Explicit expression of the square root of a quantity as its 2nd root ($\sqrt[2]{x}$)
6. Repetition of a recurring decimal digit ($\cdot\dot{m}\dot{m}$)

For further added interest, for the twelve integral powers (4, 8, 9, 16, 25, 27, 32, 36, 49, 64, 81, 100), we sought **power-clean** solutions. A solution is power-clean if it is cleanly expressed as $x^y$, where $x$ and $y$ are integers and $y$ has the least value when two or more such expressions are available (i.e., for 16, 64, and 81). For such solutions we strengthen our criteria for cleanness by proscribing the following:

7. Expression of the base $x$ as the $y$-th root of the $y$-th power of itself ($\sqrt[y]{x^y}$)

Of the 495 problems studied, we found complete solutions for 407. Of those 407 we managed to obtain a **clean sweep** (100 clean solutions, including power-clean solutions to all 12 powers) in 314 cases. Finally, of 49,500 individual solutions sought, we found 49,070—all of them clean.

Note that a specific individual problem can be denoted $n = f(i,j,k,l)$, where $n,i,j,k,$and $l$ are given and $f(i,j,k,l)$ denotes an expression sought for $n$ in terms of $i,j,k,$and $l$. For example, $59 = f(1,2,3,4)$, where $f(1,2,3,4)$ could be $4^3 - (1 \div \cdot 2)$. We refer to a number $n$ for which we could find no solution to $n = f(i,j,k,l)$ as **recalcitrant** in $(i,j,k,l)$.

## Results

We recorded our results two ways: a spreadsheet using Microsoft's Excel™, and a relational database using *Rel,* an Open Source product provided by Dave Voorhis, both included in this package. Both are as self-explanatory as reasonably possible, but we give some preliminary notes below.

We welcome corrections wherever appropriate. For example, you might find a clean or unclean solution where we have recorded a recalcitrant, or a power-clean solution we have missed.

**BeyondThe44s.zip**

This is the zipped folder you downloaded. Just unzip it to be ready to go. The folder contains:

- **Beyond The Four Fours.pdf,** this file.
- **The *Rel* database,** consisting of the folders Reldb, workspace, and Extensions (which is empty). Open the folder **BeyondThe44s** in *Rel* to work with the database.
- **Scripts**, containing a selection of *Rel* scripts relevant to this database. The script LatestBackup.rel was obtained using *Rel*'s backup facility and can be executed in *Rel* to obtain a new copy of the database.
- **Backups,** containing the three most recent *Rel* backups of the database.
- **Beyond The Four Fours.xlsx,** the spreadsheet showing the results of our study.
- **Two-digit solutions.xlsx,** a spreadsheet showing examples of identities involving just two of the nine digits. Like the authors, you may find this a useful tool in the solution of these problems. Note that examples of the form $x = x \times 1$ are shown as "unclean" but in fact clean expressions are sometimes available in such cases by judicious use of the decimal point, as in 4 = ·4 ÷ ·1, for example. Note also that the file includes a couple of charts that capture certain aspects of how the numbers from 1 to 100 can be expressed in terms of the 45 pairs of nonzero digits.

**Spreadsheet Notes** (by Gerard Joseph)

The main part of the spreadsheet is a 100 × 495 array, each row of the array corresponding to a given quartet $(i,j,k,l)$, each column to a given number $n$, and thus each cell to a given individual problem $n = f(i,j,k,l)$. The shading in each cell indicates the nature of the solution(s) found for the relevant problem, viz., at least one clean solution found, only unclean solutions found, no solution found, or at least one power-clean solution found (applicable to the powers 4, 8, 9, 16 …). The quartets themselves (one per row) are shaded somewhat correspondingly, with a special shading to indicate a clean sweep where applicable.

The shading of these cells is actually effected by conditional formatting based on the (hidden) values "R" (recalcitrant), "U" (unclean), and "P" (power-clean). The default shading, which indicates at least one clean solution, is overridden appropriately if any of those values is entered into the cell. Though logically present, those values are physically hidden (using custom cell formatting) in order to optimize the visual impact of the spreadsheet. (Note that although all the solutions we've managed to find are clean, the design of the spreadsheet continues to accommodate unclean solutions in case any are subsequently found in existing cases of recalcitrance.)

Thus shaded, the spreadsheet serves to convey a visual representation of the "topography" of the solutions we've managed to find to these problems—specifically, the incidence of recalcitrance, uncleanness, and power-cleanness across the problem space. The spreadsheet also records statistics on the three kinds of exceptional situation of interest, viz., recalcitrant numbers (no solution found), numbers for which only unclean solutions (one or more) were found, and powers for which no power-clean solution was found. Three additional columns record the total numbers of instances of each kind of exception for each quartet (*i*,*j*,*k*,*l*), while three additional rows record the corresponding totals for each number *n*. The latter sets of totals are graphically depicted in a chart.

Standard Excel filtering can be used to perform various interesting quartet-oriented queries on the solutions, e.g., finding all quartets yielding a given number of instances of recalcitrance, uncleanness, and/or (non-)power-cleanness. Similar queries relating to a given number are also possible, though due to an apparent quirk of Excel seem to require the relevant values (“P”, “U”, and “R”) to be made visible first.

**Database Notes** (by Hugh Darwen)

The database is offered as an example application of the language **Tutorial D**, devised by Hugh Darwen and Chris Date. *Rel* can be downloaded from https://reldb.org/c/.

The database records our results in three relation variables (“relvars” for short): Recalcitrant, UncleanOnly, and NonPowerClean, each having five attributes named i, j, k, l, and n, all of type INTEGER. Absence of a particular tuple <*i*,*j*,*k*,*l*,*n*> from all three relvars indicates that we found at least one clean solution for that case, power-clean in cases where *n* is a perfect integral power. (*Note:* UncleanOnly eventually became empty but was retained because it is used in many definitions of views and constraints; it also serves as a reminder of our preference for clean solutions.)

There are many other relvars, both real and virtual. The virtual variable (view) AllRelvars gives the name, explanation, and **Tutorial D** definition of each one and ConstraintsExplained does the same for each declared constraint.
A comprehensive set of views give various analyses of our results (the reader might be interested to see how these are defined in *Rel*). For example, RecalcitrantsByDigit shows the number of recalcitrants we encountered for cases containing each of the nine digits in turn. Here’s the result:

RecalcitrantsByDigit

| AllRecs | RecsOn1 | RecsOn2 | RecsOn3 | RecsOn4 | RecsOn5 | RecsOn6 | RecsOn7 | RecsOn8 | RecsOn9 |
|---|---|---|---|---|---|---|---|---|---|
| 430 | 114 | 100 | 91 | 17 | 28 | 155 | 160 | 99 | 7 |

This shows that the digit 7 is the one most likely to give rise to recalcitrance, 9 the least. And the following result (next page) analyses the effects of individual pairs of digits.

RecalcitrantsByDigitsxy

| RecsOnxy | Total | RecsOnx1 | RecsOnx2 | RecsOnx3 | RecsOnx4 | RecsOnx5 | RecsOnx6 | RecsOnx7 | RecsOnx8 | RecsOnx9 |
|---|---|---|---|---|---|---|---|---|---|---|
| RecsOn1y | 114 | 79 | 13 | 16 | 0 | 3 | 13 | 32 | 18 | 3 |
| RecsOn2y | 100 | 13 | 59 | 8 | 2 | 8 | 30 | 20 | 18 | 0 |
| RecsOn3y | 91 | 16 | 8 | 42 | 0 | 2 | 40 | 33 | 4 | 0 |
| RecsOn4y | 17 | 0 | 2 | 0 | 0 | 0 | 10 | 6 | 1 | 0 |
| RecsOn5y | 28 | 3 | 8 | 2 | 0 | 10 | 3 | 7 | 1 | 0 |
| RecsOn6y | 155 | 13 | 30 | 40 | 10 | 3 | 103 | 39 | 25 | 2 |
| RecsOn7y | 160 | 32 | 20 | 33 | 6 | 7 | 39 | 113 | 22 | 1 |
| RecsOn8y | 99 | 18 | 18 | 4 | 1 | 1 | 25 | 22 | 65 | 1 |
| RecsOn9y | 7 | 3 | 0 | 0 | 0 | 0 | 2 | 1 | 1 | 1 |

For example, the cell at the intersection of RecsOnx7 and RecsOn3y shows that of the cases involving both 3 and 7, 33 were found to be recalcitrant. Because the order of the digit operands is insignificant, the same number appears at the intersection of RecsOnx3 and RecsOn7y.