# Instructive Prime Number Reduction Algorithms

by *William J. Bruce*

*The purpose of this article is to provide teachers of elementary through high school with a collection of algorithms that are suitable as source material. These can be used either for practice with the simple arithmetical operations alone or can be used as an introduction to graphical representation, flow charts, and computer programming.*

Certain types of algorithms that utilize only prime numbers as elements for reduction are possible, but not always easy to design, let alone prove. In each one, the purpose is to design an algorithm on the set of integers N, such that $N \geq 2$, in which the first chosen operation is always division by one or more prime numbers. Also, these algorithms must reduce every integer, $N \geq 2$, to unity. Furthermore, in this reduction, they must succeed only with the inclusion of the first named operation and fail, otherwise.

The following set of such algorithms has been designed by the author, except where otherwise indicated:

1. N/2
   Odd 3N + 1
   Even N/2

2. (a) N/3
   Odd N - 1
   Even 2N + 1

   (b) N/3
   Odd (N + 1)/2
   Even 3N - 1

3. (a) N/5
   Odd 3N - 1
   Even N/2

   (b) N/5
   Odd (N + 1)/2
   Even 3N - 1

   (c) N/5
   Odd (N + 1)/2
   Even 2N + 1

   (d) N/5 N/3
   Odd N + 1
   Even 2N + 1

   If N/5 is deleted, failure occurs for N = 6.

4. (a) N/7 N/5
   Odd N - 1
   Even 2N + 1

   If N/7 is deleted, failure occurs for N = 34.

   (b) N/7 N/5
   Odd 3N + 1
   Even N/2

   If N/7 is deleted, failure occurs for N = 27.

5. N/11 N/5
   Odd 3N + 1
   Even N/2

   If N/11 is deleted, failure occurs for N = 27.

6. N/13 N/5
   Odd N - 1
   Even 2N + 1

   If N/13 is deleted, failure occurs for N = 34.

7. N/17 N/5
Odd N - 1
Even 2N + 1

If N/17 is deleted, failure occurs for N = 34.

8. N/19 N/5
Odd (N - 1)/2
Even 3N + 1

If N/19 is deleted, failure occurs for N = 4.

9. N/23 N/5
Odd N - 1
Even 2N + 1

If N/23 is deleted, failure occurs for N = 34.

10. N/29 N/11 N/5 N/3
Odd 2N - 1
Even N + 1

If N/29 is deleted, failure occurs for N = 19.
If N/11 is deleted, failure occurs for N = 31.
If N/5 is deleted, failure occurs for N = 67.
If N/3 is deleted, failure occurs for N = 21.

In each step, the operations given in the first row are to be performed first, in any order. If these operations are not possible, the separate instructions for odd and even integers are to be used, followed by another attempt to apply the operations of the first row, and so on.

For each algorithm for which deletion of the first named operation causes a failure, a sample value of N for which this failure occurs is given. In order that the algorithm be valid for the first named operation, it must fail for the other first-row operation(s) taken separately.

Algorithms 1 and 2(a) have been considered separately in two previous papers by the author. After some attempts by P. Erdös (Member of the Hungarian Academy of Science), algorithm 1 has not been proven yet. Proofs by A. Meir (Professor of Mathematics, University of Alberta) and by the author have been found for 2(a). So far, algorithms 3(c) and 3(d) are the only other ones listed for which proofs have been completed. This proof by Meir is by induction involving arithmetic mod 15 and proceeds as follows:

We want to prove that the algorithm, namely, N/5 and/or N/3, if possible,
otherwise, if N = odd → N + 1
if N = even → 2N + 1,
reduces any given integer, $n \geq 2$, to 1.

Suppose that we have proved that this theorem is true for $1 < n < m$. We shall show, then, that it is true also for $n = m$. Since 15 is the lowest common multiple of 5 and 3, we shall use m mod 15. Then, either m is divisible by 5 and/or 3 or it is off by some integer 1 to 14, inclusive. Therefore, we distinguish 15 cases according to the remainder of m mod 15 and apply the algorithm to each of these cases as follows:

1. $m = 15k + 1 <$
   - $k = \text{odd} \rightarrow 30k + 3 \rightarrow 10k + 1 < m.$
   - $k = \text{even} \rightarrow 15k + 2 \rightarrow 30k + 5 \rightarrow 6k + 1 < m.$

2. $m = 15k + 2 <$
   - $k = \text{odd} \rightarrow 15k + 3 \rightarrow 5k + 1 < m.$
   - $k = \text{even} \rightarrow 30k + 5 \rightarrow 6k + 1 < m.$

3. $m = 15k + 3 \to 5k + 1 < m.$

4. $m = 15k + 4 \begin{cases} k = \text{odd} \to 15k + 5 \to 3k + 1 < m. \\ k = \text{even} \to 30k + 9 \to 10k + 3 < m. \end{cases}$

5. $m = 15k + 5 \to 3k + 1 < m.$

6. $m = 15k + 6 \to 5k + 2 < m.$

7. $m = 15k + 7 \begin{cases} k = \text{odd} \to 30k + 15 \to 6k + 3 < m. \\ k = \text{even} \to 15k + 8 \to 30k + 17 \to 30k + 18 \to 10k + 6 < m. \end{cases}$

8. $m = 15k + 8 \begin{cases} k = \text{odd} \to 15k + 9 \to 5k + 3 < m. \\ k = \text{even} \to 30k + 17 \to 30k + 18 \to 10k + 6 < m. \end{cases}$

9. $m = 15k + 9 \to 5k + 3 < m.$

10. $m = 15k + 10 \to 3k + 2 < m.$

11. $m = 15k + 11 \begin{cases} k = \text{odd} \to 30k + 23 \to 30k + 24 \to 10k + 8 < m. \\ k = \text{even} \to 15k + 12 \to 5k + 4 < m. \end{cases}$

12. $m = 15k + 12 \to 5k + 4 < m.$

13. $m = 15k + 13 \begin{cases} k = \text{odd} \to 30k + 27 \to 10k + 9 < m. \\ k = \text{even} \to 15k + 14 \to 30k + 29 \to 30k + 30 \to 6k + 6 < m. \end{cases}$

14. $m = 15k + 14 \begin{cases} k = \text{odd} \to 15k + 15 \to 3k + 3 < m. \\ k = \text{even} \to 30k + 29 \to 30k + 30 \to 6k + 6 < m. \end{cases}$

15. $m = 15k + 15 \to 3k + 3 < m.$

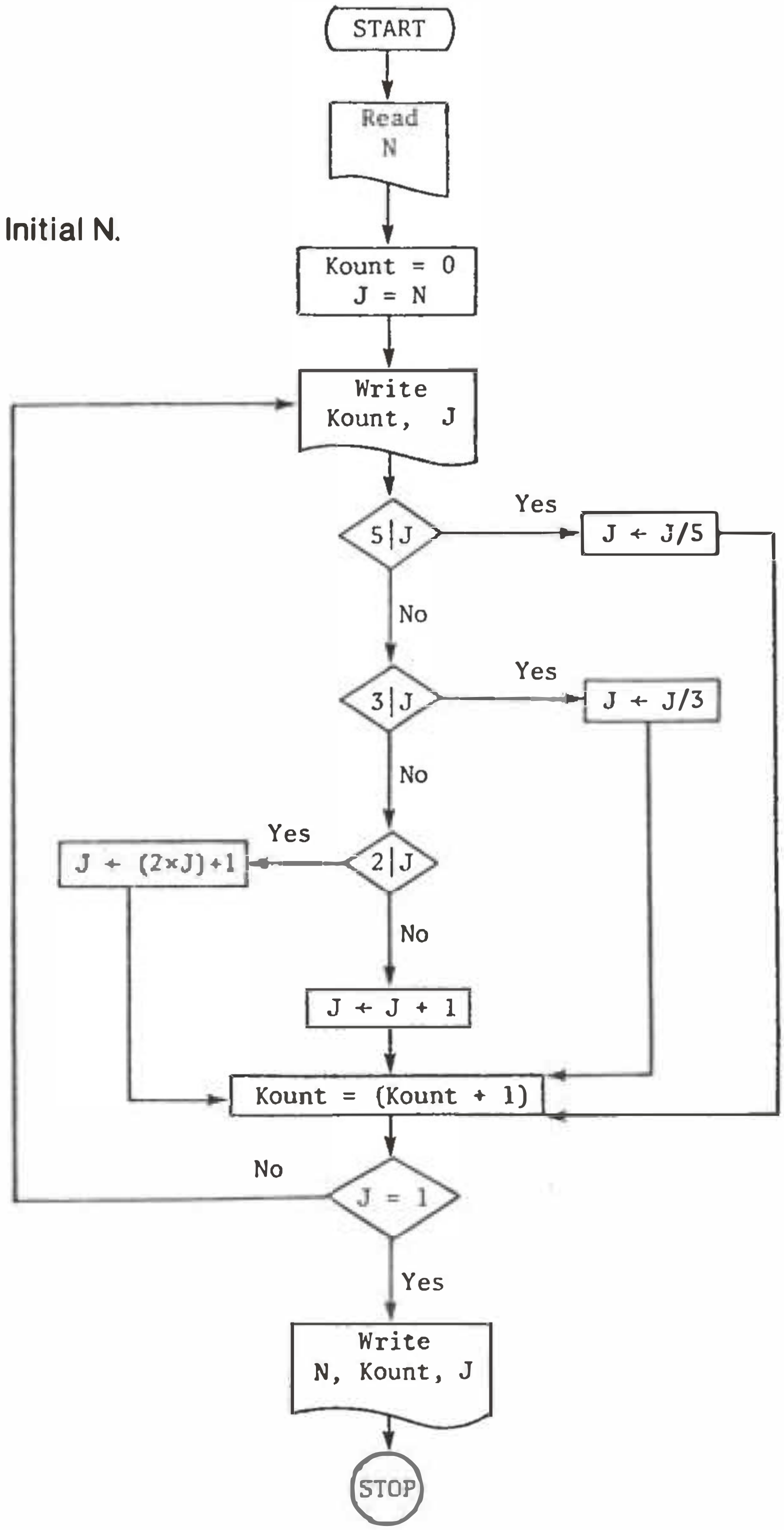
After a few steps in each case, the number obtained on the right-hand side is *smaller than the initial m*. Thus, if the theorem is true for $1 < n < m$, it is true also for $n = m$. But simple checking establishes that it is true for $n = 2$. Hence, by the principle of induction, the theorem is true for all integers $n \geq 2$.

For N = 55, the tabulation for this algorithm becomes:

| *Step Number* | *Procedure* | *n* |
|---|---|---|
| 0 | | 55 |
| 1 | n/5 | 11 |
| 2 | n + 1 | 12 |
| 3 | n/3 | 4 |
| 4 | 2n + 1 | 9 |
| 5 | n/3 | 3 |
| 6 | n/3 | 1 |

A simple form of a flow chart for algorithm 3(d) for any initial N, and based on the language of the Fortran program, might be as shown in Figure 1. With minor changes, the flow charts for the other algorithms are easily constructed. The Fortran programs are similar to those previously used by the author and require very little experience to write. All of the listed algorithms have been computer checked for 2 ≤ N ≤ 500.

FIGURE 1
Flow Chart for Chain with Initial N.



For those who want to start with a very simple prime number reduction algorithm, E. Phibbs (Professor Emeritus of Mathematics, University of Alberta) suggests:

N/5
Odd N + 1
Even 2N + ½d, where d is the last digit of the previous number obtained.

It is easy to see why this one always works. Try it.

The author has not attempted to design reduction algorithms for prime numbers greater than 29.

### References


Bruce, William J. "Crazy Roller Coasters." *The Mathematics Teacher*, Vol. 71, No. 1, January, 1978, pp.45-49.

________________. "An Instructive Algorithm Involving a Number Theoretic Problem." *Ontario Mathematics Gazette*, Vol. 17, No. 2, December, 1978, pp.44-50. (This article will be published in the February 1980 issue of *delta-k*.


---

## *Algebra Holiday Quiz*

Here is a quiz for the Christmas holiday season. U R 2 solve for the value of (x) in each of the equations below. Place your answer in the space provided at the left. When you have solved all the equations, unscramble the letters to get the message.

| | |
|---|---|
| _______ $x + M = 2M$ | _______ $bx + be = 2be$ |
| _______ $3x - s = 2s$ | _______ $2x - 2s = 0$ |
| _______ $\frac{vt^2}{g} = \frac{vxt}{g}$ | _______ $5i + x = 6i$ |
| _______ $swx = swh$ | _______ $x - b = -b + C$ |
| _______ $pr = xp$ | _______ $5x - 2r = r + 2x$ |
| _______ $\frac{x}{a} = 1$ | _______ $m^2x = m^3$ |
| _______ $7 + ax = ay + 7$ | _______ $8x + mv = 8r + mv$ |

---