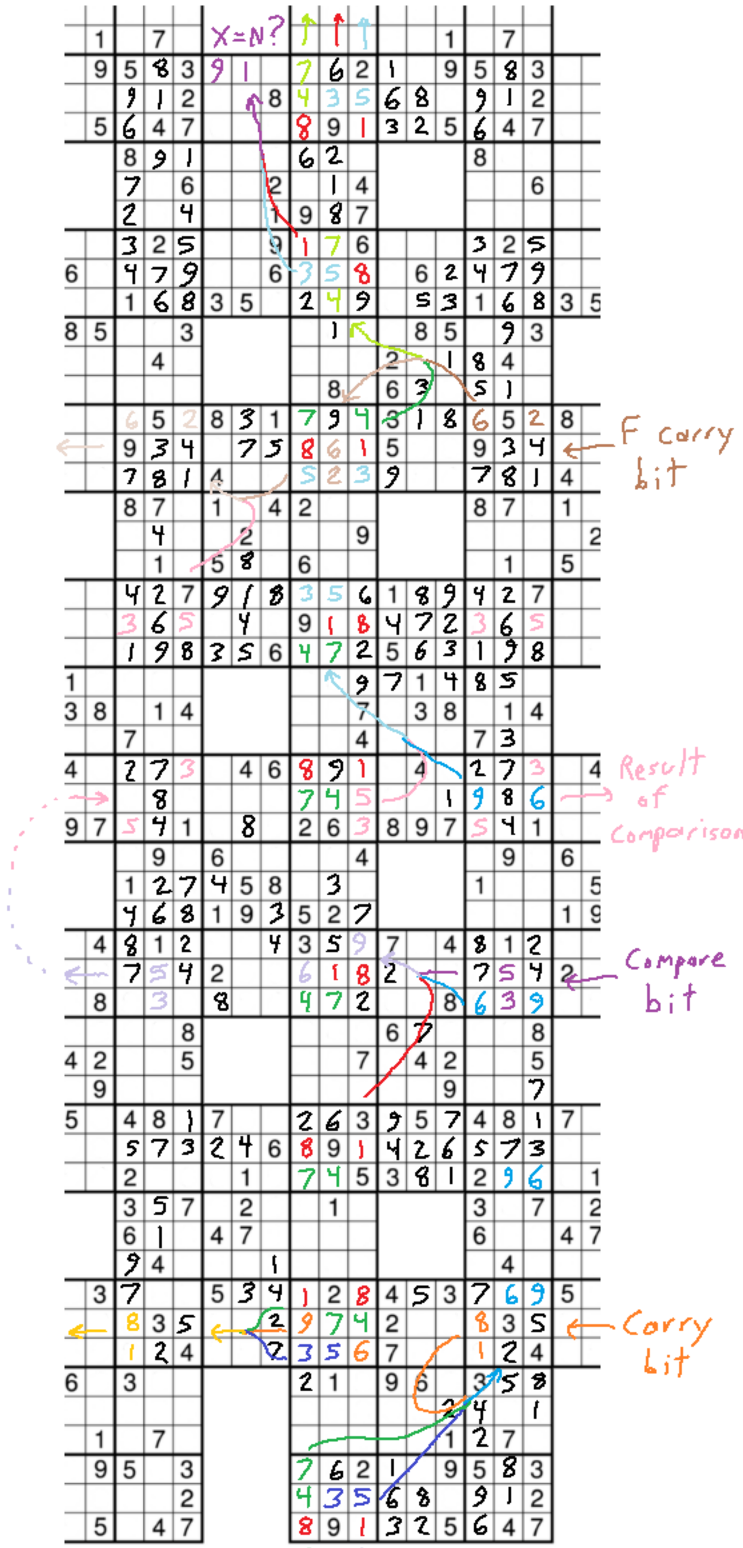

N (Number to factor)

F $\overset{?}{\rightarrow}$ F+1 (Current "factor")

X → X+F $\overset{?}{\rightarrow}$ 0 (Accumulator)

This is the world's largest Samurai Sudoku, consisting of a fairly simple pattern of givens. Each row consists of one particular Sudoku, repeated 45 or 46 times. The black givens never vary, but there are some boundary conditions given at the edges. The blue givens along the left, right, and top are simple and periodic, but the green/red givens at the bottom are a little more involved.

Solving generally proceeds from the bottom up; the complete row of green/red givens will uniquely produce a new set of green/red givens, 8 rows up. However, solving a 45+46+45+46+45+46+45+46 Sudoku is quite impractical, so experimenting with reduced versions of it (eg. 1+2+1+2+1+2+1+2 to start with) is advisable. Hopefully this will start to give some idea of how this monster behaves.

The Samurai Sudoku is in fact a simple logical machine, running a computation! In the preceding diagram, you can see black numbers where the Sudokus never vary (almost all of which can be deduced logically from neighbours), and also pairs of coloured numbers that all represent "bits" participating in the computation. A "0" bit is represented by the numbers as they are in the diagram, and a "1" bit by swapping them. I haven't written in the varying internal numbers of each Sudoku, which can be rather chaotic (because, well, Sudokus aren't really designed to be logic circuits). You can verify for yourself that these always solve uniquely and produce the desired operations, but the details aren't important (and to solve the puzzle itself you really just need enough information to make a hypothesis about what it's computing).

Two bit pairs of the same colour will propagate across a Sudoku without changing value, and I haven't explicitly written arrows showing how these bits propagate. (In most cases they're clearly linked just from placement.) Operations that actually change bits go from a darker coloured pair to a lighter one; I've added arrows showing the inputs of these operations.

At a low level, there are three numbers being acted on: N, F, and X. Each is represented by one bit per column (least significant bits on the right). F starts at 2 (hence the single swapped pair of 4 & 7 near the bottom-right of the puzzle), and X starts at 0, while N is an important input specified all along the bottom. During each solution of 8 rows, the following operations occur:

```
X <- X+F
if X > N:
  X <- 0 and F <- F+1
if X = N:
  stop (optionally)
```

So at a high level, the Sudoku is factoring the input number N. There is exactly one solution to the Sudoku for each factor of N except 1. (Technical note: Once F becomes greater than N, it just keeps incrementing until it wraps around to 0, at which point the values never change again.)

Since we're looking for the solution of minimal height (as specified in the puzzle), we need to know the smallest proper factor of N. N's binary representation, given at the bottom of the puzzle, is 011100111001011010010111001100011011100111111, or 15,886,327,363,391 in decimal. Wolfram Alpha will tell you this factors into the primes 1,701,181 * 9,338,411. The total number of N,F,X rows is (N/2+1) + (N/3+1) + … (N/1701181+1), where / represents integer division. This is simple to compute using a separate program: 221202005182579. This corresponds to 221202005182576 replications of the inner section, matching the ___76 specified. The rest of this number splits into 2, 21, 20, 20, 05, 18, and 25; the final answer to the puzzle is BUTTERY. Whew!