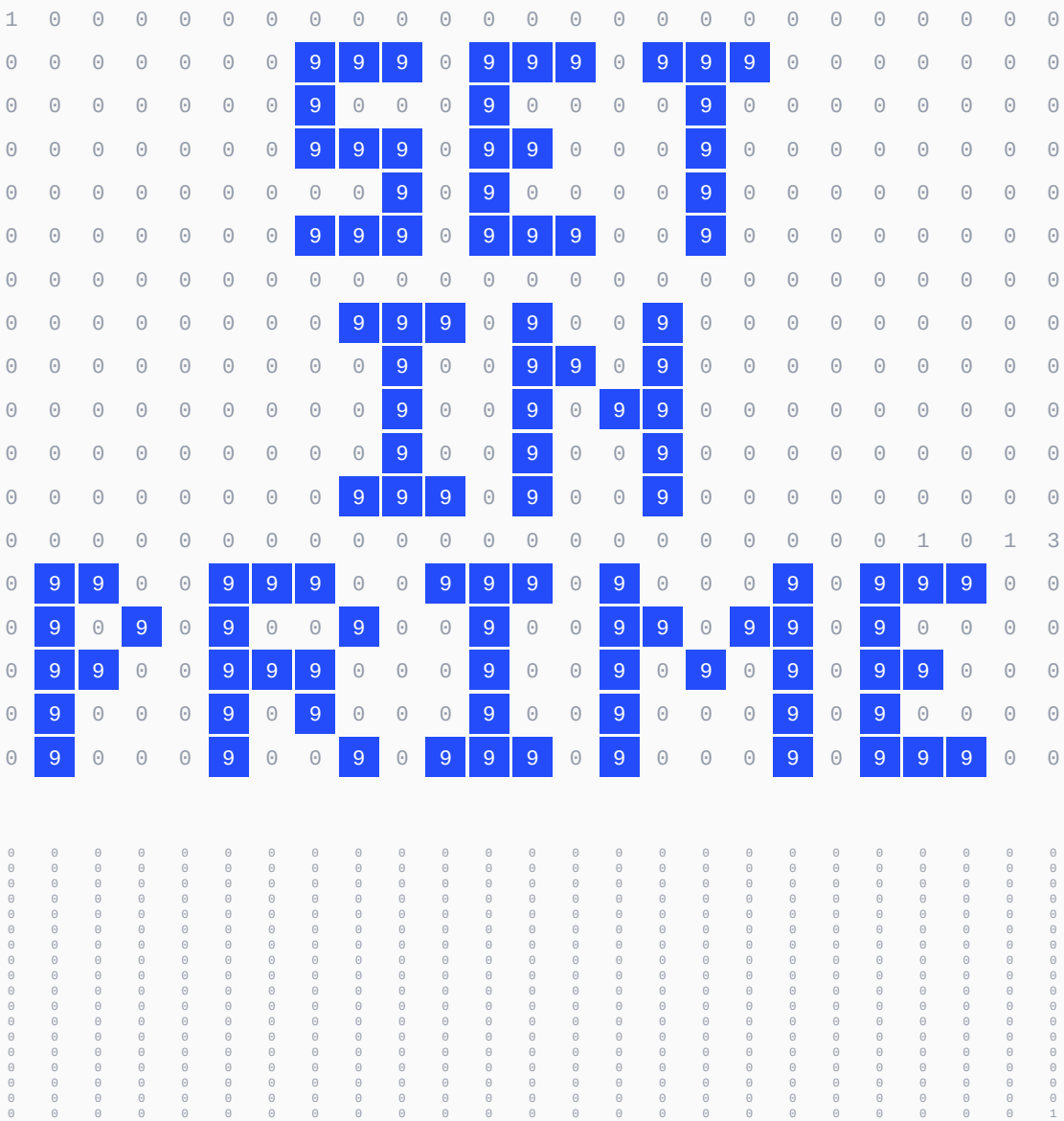
SET IN PRIME

set in prime

A name, written into a prime.



For the joy of discovery.

900 DIGITS / VERIFIED WITH LEAN

Every digit belongs. Read left to right, then top to bottom.

6b50b6e82b1d635b9ca55f1a

set in prime

WHAT IS TRUE

The 900-digit number in this portrait is prime. Its only positive divisors are 1 and itself. Lean also proves that the exact ordered list of decimal digits reconstructs this number.

CHECKED BY LEAN

Lean is software for checking mathematical proofs. It checks that a conclusion follows from the definitions and rules of mathematics. For this portrait, it checks an exact argument that the number is prime, rather than a guess or a sample of possible divisors.

HOW TO READ IT

Read every digit from left to right, then top to bottom, including the smaller zero block. Blue tiles spell SET / IN / PRIME. Six reserved background cells contain 001013; the search adjusts these cells without changing the letters.

WHY THE PROOF WORKS

The certificate proves that any prime factor would have to be larger than the square root of this number. Every composite number has a prime factor at or below its square root. Therefore, this number is prime.

The known factor 10^450 of N - 1 and witness 11 make this Pocklington-style argument practical. The lower zeros are part of the number and its construction.

CHECK IT YOURSELF

The next page shows a piece of your actual proof. Download proof.zip beside this portrait for the complete source, exact number, verification log and instructions. Anyone can check those files with Lean; no paid API or account is required.

The SVG digit order and positions are audited separately. Recognizing the name, rendered pixels, ownership and novelty are not claims made by the Lean proof.

6b50b6e82b1d635b9ca55f1ac70c52f67e47b081d4b4c061baacd4b899f8c72f

SET IN PRIME / INSIDE YOUR PROOF

A proof you can inspect.

A large number can look prime and still have hidden factors. A mathematical proof settles the question. Lean checks the written argument precisely, and its small core checker verifies the proof before accepting the result.

FROM YOUR ACTUAL LEAN FILE

Here N means the exact number printed in your portrait. Nat.Prime N is the statement that this number is prime. The lines below begin its proof by applying a proven mathematical rule.

```
theorem is_prime : Nat.Prime N := by
  unfold N
  apply PrimeCriterion.prime_of_decimal_pocklington
    (r := R) (a := 11) (k := 450) (by decide) (by decide)

theorem digits_count : digits.length = 900 := by decide
```

These are exact excerpts from GeneratedPrime.lean, not the full proof. The intervening proof steps are in the download. The final line above also checks how many digits the portrait contains.

WHAT HAPPENS WHEN IT RUNS?

Lean reads the file, checks the calculations and verifies that the proof follows the rules. It either accepts the proof or reports an error. It is checking a mathematical argument, not asking an AI whether the answer looks right.

```
lake env lean --trust=0 GeneratedPrime.lean
```

This is the check command after installing the pinned dependencies in README.md. Running the downloadable project yourself lets you check the proof without relying on the Set in Prime website.

THIS PORTRAIT'S CHECK

Accepted by Lean 4.30.0 in 19.83 seconds, with trust level zero. The axiom audit allows only Lean's standard logical axioms. No admitted proof or native_decide is used.

VERIFIED LEAN SOURCE / SHA-256

8496a9b81bb03479638e657766dcf4b9f8862809d141c1dd35c0bcff8e1a3d3a

6b50b6e82b1d635b9ca55f1ac70c52f67e47b081d4b4c061baacd4b899f8c72f