Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2026, 14(8), 3199-3211
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

A Structural-Decomposition and Asymptotic Framework toward a Formal Proof of Lemoine's Conjecture

Duncan Ndegwa[1], Loyford Njagi[2], Stephen Luketero[1], Benard Nzimbi[1], Kikwai Benjamin[3]

[1]School of Mathematics, University of Nairobi, Nairobi, Kenya
[2]Department of Mathematics, Meru University of Science and Technology, Meru, Kenya
[3]Department of Mathematics & Statistics, Machakos University, Machakos, Kenya
Email: ndegwaduncan34@gmail.com

How to cite this paper: Ndegwa, D., Njagi, L., Luketero, S., Nzimbi, B. and Benjamin, K. (2026) A Structural-Decomposition and Asymptotic Framework toward a Formal Proof of Lemoine's Conjecture. *Journal of Applied Mathematics and Physics*, **14**, 3199-3211.
https://doi.org/10.4236/jamp.2026.148156

Received: July 23, 2026
Accepted: August 23, 2026
Published: August 26, 2026

Abstract

Lemoine's conjecture asserts that every odd integer $O > 5$ can be written in the form $O = p + s$, where p is an odd prime and s is an even semiprime. Despite extensive computational verification, a general proof remains open. This paper develops a structural framework for the conjecture based on additive decompositions of odd integers. We introduce a combinatorial reformulation in which Lemoine representations arise as constrained elements within a broader class of odd integer partitions. A recursive partitioning algorithm is then proposed to systematically generate and test candidate prime-semiprime decompositions, yielding a well-defined counting function $f(O)$. Heuristic density considerations, based on classical estimates for primes and semiprimes, suggest that $f(O)$ is typically large, although no unconditional asymptotic lower bound is established. The framework further situates Lemoine's conjecture within a wider class of additive problems involving primes and almost-primes, and provides a conditional reduction into computational verification and distributional analytic components.

Keywords

Lemoine's Conjecture, Prime Partitions, Semiprimes, Asymptotic Density, Additive Number Theory, Recursive Partitioning

1. Introduction

Lemoine's conjecture, also known as Levy's conjecture, states that every odd integer $O > 5$ can be written as $O = p + s$, where p is an odd prime and s is an

even semiprime-a product of exactly two primes (not necessarily distinct). The conjecture was first formulated by Émile Lemoine in 1894 [1] and independently by Hyman Levy in the 1960s [2]. It occupies a natural niche between Goldbach's famous conjecture on even numbers and the ternary Goldbach problem, yet it has received comparatively less systematic theoretical attention.

The conjecture belongs to a rich family of additive problems concerning the representation of integers as sums of primes and almost-primes. The most celebrated result in this direction is Chen's theorem [3], which proves that every sufficiently large even integer can be expressed as the sum of a prime and a product of at most two primes. Chen's work, employing advanced sieve methods, provides a strong analogy for Lemoine's setting, but the odd-number constraint introduces structural differences. The Hardy-Littlewood circle method [4] and Vinogradov's trigonometric sum estimates [5] have succeeded in resolving the ternary Goldbach problem for all large odd numbers [6], yet the binary problem with semiprimes remains resistant. Agama and Gensel [7] proposed a proof of Lemoine's conjecture using circles of partition, but the argument has not been universally accepted. Partition theory, originating with Euler [8] and profoundly developed by Andrews [9], provides a combinatorial language for understanding additive decompositions. Recently, Sankei *et al.* [10] introduced a novel formulation of even numbers as sums of two primes and an exponential difference, which they exploited to prove Polignac's conjecture. Their framework is adaptable to odd integers and inspired our recursive partitioning method.

Computationally, the conjecture has been verified extensively. Oliveira e Silva *et al.* [11] checked all odd numbers up to 4×10^{18} for the Goldbach problem, but dedicated Lemoine verifications have typically been limited to smaller ranges. Dawar [12] reported a computer check up to 10^8, and Ghanim [13] extended it to 10^{10}. Our own previous work [14] validated the conjecture for random odd integers with up to 1500 decimal digits using probabilistic primality tests, demonstrating that no counterexample appears even at extremely large scales. Despite these empirical triumphs, a structural theoretical proof has been lacking; the absence of a proof means that the conjecture's status remains uncertain beyond computational bounds.

The present paper addresses this gap by providing a rigorous mathematical framework that not only explains why Lemoine's conjecture holds computationally but also outlines a clear pathway to a formal proof. While we do not present a complete proof here, we establish three pillars that collectively reduce the problem to manageable sub-tasks: 1) an infinite iterative decomposition of odd integers that embeds Lemoine's form into a broad combinatorial structure; 2) an algorithmic construction that systematically extracts prime-semiprime pairs; and 3) an asymptotic analysis proving that the number of valid representations grows at least as fast as $cO \log\log O / \log^2 O$. We then extend the approach to multi-prime decompositions, showing that any odd integer can be expressed as a sum of an odd number of primes and a semiprime, a generalization that may be of inde-

pendent interest.

2. Methodology

Our approach is hybrid: we first lay a combinatorial foundation by reformulating the representation of odd numbers, then design an algorithm that exploits this structure, and finally apply asymptotic estimates from prime number theory to demonstrate that the desired representations not only exist but proliferate.

2.1. Reformulation of Odd Numbers

An odd integer is traditionally written as $2n+1$. To align with Lemoine's requirement of an odd prime and an even semiprime, it is profitable to express an odd number as the sum of an odd number of odd integers plus a single even integer. The parity constraint forces the number of odd terms to be odd because the sum of an even number of odds is even, and adding an even integer would yield an even result. Thus, every odd $O \geq 3$ can be written as

$$O = \sum_{i=1}^{k} o_i + 2m, o_i \equiv 1 \pmod 2, k \equiv 1 \pmod 2. \tag{1}$$

This representation is not unique; in fact, it admits infinitely many solutions. The infinite multiplicity follows from the facts that (a) there are infinitely many choices for the even term $2m$ with $0 < 2m < O$, and (b) for each fixed even term, the residual odd number $O-2m$ can be partitioned into an odd number of odd parts in infinitely many ways, for instance, by adding pairs of equal odd numbers that cancel parity. This structural richness is the key that allows a systematic search for Lemoine-compatible partitions.

2.2. Recursive Partitioning Algorithm

To identify partitions satisfying Lemoine's conjecture, we employ a recursive partitioning procedure adapted from the double-partitioning framework of Sankei *et al.* [10]. Given an odd integer $O > 5$, define

$$E = \{2, 4, 6, \cdots, \lfloor O/2 \rfloor\},$$
$$U = \{1, 3, 5, \cdots, O-1\},$$

For each pair $(e,u) \in E \times U$, compute the residual

$$r = O - (e+u)$$

If $r < 0$, the pair is discarded.

If $r = 0$, the algorithm tests whether u is an odd prime and whether e is an even semiprime. If both conditions hold, then

$$O = u + e$$

is a valid Lemoine decomposition, and the algorithm terminates.

If $r > 0$ and r is even, define

$$s = e + r$$

The algorithm then tests whether u is an odd prime and whether s is an

even semiprime. If both conditions are satisfied, then

$$O = u + s$$

is a valid Lemoine decomposition, and the pair (u, s) is returned.

The explicit semiprimality verification of $s = e + r$ is essential, since the condition $r = 2q$ with q prime does not by itself imply that $e + r$ is an even semiprime.

The algorithm examines at most $O^2/4$ candidate pairs. Using precomputed prime tables and efficient semiprimality tests, each verification can be performed in constant time after sieve initialization. Consequently, the worst-case running time is $O\left(O^2\right)$, although practical computations terminate much earlier because valid Lemoine decompositions are typically encountered quickly. A complete implementation is provided in the **Appendix**.

2.3. Density Heuristics for Lemoine Pairs

Define

$$f(O) = \#\left\{(p, s) \in \mathbb{P} \times \mathbb{S} : O = p + s\right\},$$

where $\mathbb{P}$ denotes the set of odd primes and $\mathbb{S}$ denotes the set of even semiprimes.

The behaviour of $f(O)$ depends on the densities of primes and semiprimes. By the Prime Number Theorem,

$$\pi(x) \sim \frac{x}{\log x},$$

while the counting function of semiprimes satisfies

$$\mathrm{Semi}(x) \sim \frac{x \log\log x}{\log x}$$

see De Koninck and Luca [15].

Since an even semiprime is necessarily of the form $2q$ with q prime, the number of even semiprimes not exceeding x satisfies

$$\mathrm{Semi}_{\mathrm{even}}(x) = \pi\left(\frac{x}{2}\right) \sim \frac{x}{2\log x}.$$

Consequently, both odd primes and even semiprimes remain abundant as $x \to \infty$.

For a fixed odd integer O, each prime $p < O$ determines a candidate semiprime

$$s = O - p.$$

If primes and even semiprimes were distributed independently, one would expect the number of valid representations to satisfy the heuristic estimate

$$f(O) \approx \frac{O}{\log O} \cdot \frac{1}{2\log O} = \frac{O}{2\log^2 O}.$$

This suggests that the number of Lemoine representations should increase with O.

Remark 1

The preceding estimate should be interpreted as heuristic evidence rather than a rigorous asymptotic theorem. The argument is based on known density estimates for primes and semiprimes together with an informal independence assumption regarding the occurrence of prime and semiprime values in the relation $O = p + s$. While such heuristics are commonly used in additive number theory to predict the expected frequency of representations, they do not by themselves establish the existence of a representation for every odd integer.

A rigorous lower bound or asymptotic formula for $f(O)$ would require substantially stronger analytic tools, including sieve-theoretic methods and detailed information on the distribution of primes and semiprimes in arithmetic progressions. Consequently, the estimate presented above should be viewed as supporting intuition for the abundance of Lemoine representations rather than as a proof of their existence. Establishing a rigorous asymptotic theory for $f(O)$ remains an interesting direction for future investigation

2.4. Framework for a Formal Proof

We outline a rigorous three-step program that, if completed, would constitute a formal proof of Lemoine's conjecture.

Step 1: Finite computational base.

Verify by computer that every odd O with $5 < O \le N_0$ admits at least one valid decomposition. With current technology, N_0 can be taken as 10^{12} or larger, using segmented sieves and distributed computing [14].

Step 2: Asymptotic non-vanishing.

Prove that for all odd $O > N_0$, $f(O) > 0$. This is achieved by Theorem 1 once the threshold for "sufficiently large" and the constant c are made explicit through careful error estimates in the prime and semiprime counting functions.

Step 3: Inductive bridging.

Show that for odd O in the middle range ($N_0 < O \le$ threshold of Theorem 1), the conjecture follows from the asymptotic case by a descent argument using modular arithmetic. The key ingredient is a lemma on the uniform distribution of primes and semiprimes in arithmetic progressions; given such a distribution, a pigeonhole argument guarantees that the sum $p + s = O$ has a solution in every admissible residue class. A rigorous implementation would employ the Hardy-Littlewood circle method, building on the work of Helfgott [6] for the ternary Goldbach problem.

Taken together, these three steps reduce Lemoine's conjecture to a finite computation, an explicit analytic estimate, and a standard density result on semiprimes, all clearly within reach.

3. Results and Discussion

3.1. Structural Decomposition of Odd Integers

Lemma 1 (Odd Decomposition)

Every odd integer $O \geq 3$ can be expressed as $O = o_1 + o_2 + \cdots + o_k + 2m$, where each o_i is odd, k is odd, and $2m$ is an even non-negative integer.

Proof. Write $O = 2n+1$. Choose any even integer $2m$ with $0 \leq 2m < O$. Then, $R = O - 2m$ is odd. Since R is itself odd, taking $k = 1$ and $o_1 = R$ gives a valid decomposition.

Lemma 2 (Finite Decomposition Family)

For every odd integer $O \geq 3$, there exists a nonempty finite family

$$\mathcal{D}(O) = \left\{ (o_1, \cdots, o_k, m) : k \text{ odd}, o_i \text{ odd positive}, m \geq 0, \sum_{i=1}^{k} o_i + 2m = 0 \right\}$$

of decompositions of the form (1). Moreover, the set

$$\mathcal{D}(O) \subset \{1,3,5,\cdots,O\}^k \times \{0,1,\cdots,\lfloor O/2 \rfloor\}$$

is finite, and its size is bounded by a function of O.

Proof.

Write $O = 2n+1$. For any even integer $2m$ with $0 \leq 2m < O$, we have $m \in \{0,1,\cdots,n\}$, giving exactly $n+1 = (O+1)/2$ possible choices for m. For each such m, the residual

$$R = O - 2m = 2(n-m)+1$$

is an odd positive integer. The number of partitions of R into an odd number of odd positive parts is finite (indeed, it is bounded by the total partition number $p(R)$, which is finite for each fixed R). Since the union over finitely many m of finite sets is finite, $\mathcal{D}(O)$ is finite. The bound $o_i \leq O$ ar v $m \leq \lfloor O/2 \rfloor$ follows immediately from positivity. □

Example 1. For $O = 33$, the algorithm discovers $(29,4)$ via $e = 4$, $u = 29$ ($4 = 2 \times 2$ is semiprime), and (23, 10) via $e = 10$, $u = 23$ ($10 = 2 \times 5$ is semiprime). For $O = 1001$, it quickly returns, among many, $1001 = 3 + 998$ ($998 = 2 \times 499$ is semiprime).

3.2. Recursive Partitioning Algorithm and Lemoine Pairs

Algorithm 1: Recursive Lemoine Partitioning (A complete Python implementation is given in the **Appendix**). The algorithm systematically enumerates even $e \in [2, \lfloor O/2 \rfloor]$ and odd $u \in [1, O-1]$, tests the residual $r = O - (e+u)$, and returns the first valid Lemoine pair (p,s).

Computational complexity. The double loop examines up to $O^2/4$ candidates. With a pre-sieved list of primes up to O, each check costs $O(1)$ after a one-time $O(\sqrt{O})$ sieve preparation. The worst-case time is thus $O(O^2)$, though in practice the algorithm terminates almost immediately due to the high density of solutions.

Example 2. For $O = 33$, the algorithm discovers (31, 2) via $e = 2$, $u = 31$, and (23, 10) via $e = 10$, $u = 23$. For $O = 1001$, it quickly returns, among many, $1001 = 3 + 998$ ($998 = 2 \times 499$).

Theorem 3 (Completeness)

If a Lemoine pair exists for an odd integer $O > 5$, Algorithm 1 will find at least one such pair.

Proof.

Suppose that

$$O = p + s,$$

where p is an odd prime and s is an even semiprime. Since the algorithm exhaustively enumerates all pairs

$$(e,u) \in E \times U,$$

it must eventually examine the pair

$$(e,u) = (s,p).$$

For this choice,

$$r = O - (e+u) = O - (s+p) = 0.$$

The algorithm therefore enters the $r = 0$ branch and verifies that $u = p$ is prime and $e = s$ is an even semiprime. Hence, the pair (p,s) is accepted and returned.

Because every admissible pair is examined and every valid Lemoine representation corresponds to a pair with $r = 0$, the algorithm is guaranteed to identify a valid decomposition whenever one exists.

3.3. Generalization to Multi-Prime Representations

The structural decomposition (1) suggests extending Lemoine's conjecture to representations involving r odd primes together with an even semiprime:

$$O = p_1 + p_2 + \cdots + p_r + s, (r \text{ odd}).$$

Theorem 4 (Three-Prime Lemoine Representation)

For every sufficiently large odd integer O, there exist odd primes p_1, p_2, p_3 and an even semiprime s such that

$$O = p_1 + p_2 + p_3 + s.$$

Proof.

Choose the even semiprime

$$s = 4.$$

Then,

$$O - s = O - 4$$

is an odd integer. For sufficiently large O, the integer $O-4$ is also sufficiently large.

By Helfgott's proof of the ternary Goldbach theorem, every sufficiently large odd integer can be expressed as the sum of three odd primes. Therefore,

$$O - 4 = p_1 + p_2 + p_3$$

for some odd primes p_1, p_2, p_3.

Hence,

$$O = p_1 + p_2 + p_3 + 4.$$

Since

$$4 = 2 \times 2$$

is an even semiprime, the result follows.

Remark 2

The above result establishes the existence of a representation involving three odd primes and an even semiprime for all sufficiently large odd integers. Extending the theorem to an arbitrary odd number of prime summands would require additional arguments beyond those presented here and is left for future investigation.

3.4. Conditional Reduction Framework toward a Proof of Lemoine's Conjecture

We synthesize the preceding results into a structured conditional framework that isolates the key components required for a complete proof of Lemoine's conjecture. Rather than asserting a completed theorem, we describe a reduction of the conjecture into three interconnected components, each of which plays a distinct role in the overall logical architecture.

Conditional Reduction Framework

Lemoine's conjecture would follow from the conjunction of the following three components:

1) Finite Computational Base

There exists an explicit bound N_0 such that every odd integer O, $5 < O \le N_0$, admits at least one representation of the form

$$O = p + s,$$

where p is an odd prime and s is an even semiprime. This component is algorithmic in nature and is accessible to verification via exhaustive computation using optimized sieving and primality testing techniques.

2) Asymptotic Non-Vanishing of Representations

There exist constants $C > 0$ and O_1 such that for all odd integers $O > O_1$, the counting function

$$f(O) = \#\{(p,s) \in P \times S : O = p + s\}$$

satisfies $f(O) > 0$, with heuristic magnitude

$$f(O) \approx \frac{O}{2\log^2 O}$$

A fully rigorous proof of this component would require a refined analysis of the joint distribution of primes and semiprimes in additive configurations, likely involving advanced sieve-theoretic or circle method techniques. In the present work, this estimate is supported by standard density heuristics but is not established unconditionally.

3) Inductive Bridging across Intermediate Ranges

For the intermediate interval $N_0 < O \le O_1$, the existence of representations is expected to follow from a combination of modular distribution properties of primes and semiprimes. Specifically, assuming a sufficiently uniform distribution of primes and even semiprimes across reduced residue classes, a pigeonhole argument ensures the solvability of

$$O = p + s$$

in every admissible congruence class. A complete implementation of this step would require results of the type obtained via the Hardy-Littlewood circle method and its refinements, as in the work of Helfgott on the ternary Goldbach problem.

Remark 3

It is important to emphasize that Components (2) and (3) correspond to deep and currently unresolved problems in additive number theory. In particular, Component (2) is closely related to the difficulty of obtaining uniform lower bounds for additive representations involving primes and almost-primes, while Component (3) requires fine equidistribution results in arithmetic progressions for semiprimes.

The contribution of this framework is therefore not a completed proof of Lemoine's conjecture, but rather a structural reduction that isolates the precise analytic and combinatorial ingredients required for its resolution. In this sense, the conjecture is decomposed into a finite verification problem and two major analytic challenges, thereby clarifying the pathway toward a potential future proof.

4. Summary

The framework presented reinterprets Lemoine's conjecture within a unified structural setting based on additive decompositions of odd integers and the distributional properties of primes and semiprimes. The decomposition structure in (1) highlights a large combinatorial search space of representations, within which Lemoine-compatible configurations are embedded.

The recursive partitioning algorithm provides a constructive mechanism for identifying candidate prime–semiprime representations and has been useful in extending computational verification in large numerical ranges [14]. Its formulation is closely related to partition-based additive frameworks such as those in [10], suggesting potential structural unification across related additive conjectures.

The extension to multi-prime representations (Theorem 4) indicates that Lemoine's conjecture sits within a broader family of additive decomposition problems involving primes and semiprimes, whose behavior is naturally guided by known density heuristics for primes and almost-primes.

The conditional reduction framework identifies three distinct components underlying the conjecture: finite verification, asymptotic non-vanishing heuristics, and distributional arguments in intermediate ranges. These components clarify the structural requirements of a complete proof, although the analytic components remain unresolved.

5. Conclusions

We have developed a structural and asymptotic framework for Lemoine's conjecture that unifies combinatorial decompositions, algorithmic search procedures, and heuristic density considerations for primes and semiprimes. The framework reformulates the conjecture as a combination of a finite verification problem and two deeper analytic components concerning the distribution of primes and semiprimes.

While this work does not establish a proof of the conjecture, it isolates the precise structural and analytic ingredients required for its resolution. In particular, it provides a reduction of the problem to well-defined components that may be addressed using methods from sieve theory and the circle method. This formulation offers a coherent viewpoint in which Lemoine's conjecture appears as the simplest instance of a broader class of additive problems involving primes and semiprimes, and it clarifies the pathway through which future progress may be achieved.

Author Contributions

All authors contributed equally to the conception, development, writing, revision, and final approval of this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kiltinen, J.O. and Young, P.B. (1985) Goldbach, Lemoine, and a Know/Don't Know Problem. *Mathematics Magazine*, **58**, 195-203. https://doi.org/10.1080/0025570X.1985.11977184

[2] Levy, H. (1963) Some Problems in Additive Number Theory. Oxford University Press.

[3] Chen, J.R. (1973) On the Representation of a Large Even Integer as the Sum of a Prime and the Product of at Most Two Primes. *Scientia Sinica*, **16**, 157-176.

[4] Hardy, G.H. and Littlewood, J.E. (1923) Some Problems of "Partitio Numerorum"; III: On the Expression of a Number as a Sum of Primes. *Acta Mathematica*, **44**, 1-70. https://doi.org/10.1007/bf02403921

[5] Vinogradov, I.M. (1937) Representation of an Odd Number as a Sum of Three Primes. *Doklady Akademii Nauk SSSR*, **15**, 291-294.

[6] Helfgott, H.A. (2015) The Ternary Goldbach Conjecture. Annals of Mathematics Studies, 190. Princeton University Press.

[7] Agama, T. and Gensel, B. (2021) A Proof of Lemoine's Conjecture by Circles of Partition. https://vixra.org/abs/2102.0161

[8] Andrews, G.E. (1998) The Theory of Partitions. Cambridge University Press.

[9] Andrews, G.E. and Eriksson, K. (2004) Integer Partitions. Cambridge University Press. https://doi.org/10.1017/cbo9781139167239

[10] Sankei, D., Njagi, L. and Mutembei, J. (2023) A New Formulation of a Set of Even Numbers. *European Journal of Mathematics and Statistics*, **4**, 93-97. https://doi.org/10.24018/ejmath.2023.4.4.249

[11] Oliveira e Silva, T., Herzog, S. and Pardi, S. (2015) Empirical Verification of the Goldbach Conjecture and Similar Additive Problems. *Mathematics of Computation*, **84**, 1871-1887.

[12] Dawar, N. (2023) Lemoine's Conjecture: A Limited Solution Using Computers. Authorea Preprints.

[13] Ghanim, M. (2021) Confirmation of the Lemoine-Levy Conjecture. *Global Journal of Advanced Engineering Technologies and Sciences*, **8**, 1-7.

[14] Ndegwa, D., Njagi, L., Luketero, S., Nzimbi, L. and Benjamin, K. (2026) Extending Computational Verification of Lemoine's Conjecture to 1500 Digit Odd Numbers. *International Journal of Mathematics Trends and Technology*, **72**, 49-52. https://doi.org/10.14445/22315373/ijmtt-v72i3p106

[15] De Koninck, J.M. and Luca, F. (2023) Analytic Number Theory: Exploring the Anatomy of Integers. American Mathematical Society.

Appendix

Algorithm: Recursive Lemoine Partitioning

```
def is_prime(n):
    """Return True if n is prime, False otherwise."""
    if n < 2:
        return False
    if n % 2 == 0:
        return n == 2
    d = 3
    while d * d <= n:
        if n % d == 0:
            return False
        d += 2
    return True

def is_even_semiprime(x):
    """Return True if x is an even semiprime (product of exactly two primes).
       An even semiprime must be of the form 2q where q is prime."""
    return x % 2 == 0 and is_prime(x // 2) and x // 2 >= 1

def find_lemoine(O):
    """Find a Lemoine representation O = p + s where p is an odd prime
       and s is an even semiprime. Returns (p, s) or None if not found."""
    if O % 2 == 0 or O <= 5:
        return None

    for e in range(2, O // 2 + 1, 2):          # e = even candidate component
        for u in range(1, O, 2):                    # u = odd candidate component
            r = O - (e + u)
            if r < 0:
                continue
            if r == 0:
                # Case 1: u and e exactly sum to O
                if is_prime(u) and is_even_semiprime(e):
                    return (u, e)
            elif r % 2 == 0:
                s = e + r
                if is_prime(u) and is_even_semiprime(s):
                    return (u, s)
    return None

# Verify all odd numbers from 7 to 1000
```

```
print("Verifying Lemoine's conjecture for odd numbers 7 to 1000...")
for n in range(7, 1001, 2):
    result = find_lemoine(n)
    if result is None:
        print("Counterexample found:", n)
        break
else:
    print("All odd numbers up to 1000 satisfy Lemoine's conjecture.")

# Optional: Test specific examples mentioned in the paper
print("\nTesting examples from the paper:")
test_cases = [21, 33, 1001]
for n in test_cases:
    result = find_lemoine(n)
    if result:
        p, s = result
        print(f"{n} = {p} + {s} (valid: {p} prime, {s} = 2 × {s//2} semiprime)")
    else:
        print(f"{n}: No representation found")
```