Ganitam: Frontier Journal of Computational and Applied Mathematics
Vol. 1 No. 2 (December, 2026)

Received: 10th June 2026
Revised: 16th August 2026
Accepted: 19th September 2026

Licensing Statement: This article is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0) (https://creativecommons.org/licenses/by/4.0/).[11]

A COMPUTATIONAL STUDY OF THE FREQUENCY OF FERMAT PSEUDOPRIMES

SHUBHAM AGARWAL[1#], BHAWANA FULARA[2], DEEPAK KUMAR SHARMA[3], ARVIND BHATT[4]

ABSTRACT. Fermat pseudoprimes constitute an important class of composite integers that satisfy a congruence associated with Fermat's little theorem and consequently cannot be distinguished from primes by a single Fermat primality test using a particular base. Their occurrence is significant in computational number theory because it illustrates the limitations of elementary primality-testing procedures and motivates the development of more reliable algorithms. This paper investigates the frequency of Fermat pseudoprimes through their mathematical characterization, computational detection, and distribution across increasing numerical ranges. For an integer base $a \geq 2$, a composite integer n relatively prime to a is a Fermat pseudoprime to base a if $a^{(n-1)} \equiv 1 \pmod{n}$. The study focuses primarily on base 2, while establishing a framework for comparisons across different bases.

The computational methodology uses modular exponentiation, composite-number filtering, cumulative counting, and normalized frequency measurements. Published computational data for base-2 Fermat pseudoprimes are used to establish reference benchmarks, including counts of 5,597 pseudoprimes below 10^9, 14,884 below 10^{10}, and 101,629 below 10^{12}. These benchmarks demonstrate that the absolute number of pseudoprimes increases with the search bound, although their proportion among all integers decreases. The paper also discusses the influence of the testing base, the significance of Carmichael numbers, computational complexity, and the distinction between empirical observations and asymptotic results. A reproducible algorithm is presented to facilitate independent verification and extension of the investigation. The study provides a framework for examining pseudoprime frequency and assessing the limitations of Fermat-based primality testing.

1. Introduction

Prime numbers are fundamental objects in number theory and play a central role in modern computational mathematics. Their applications extend to cryptography, coding theory, algorithm design, and computational algebra. Determining whether a sufficiently large integer is prime is therefore an important computational problem. Although several deterministic and probabilistic primality-testing algorithms are available, elementary tests based on modular arithmetic remain valuable for understanding the mathematical foundations of primality detection (Crandall & Pomerance, 2005). Fermat's little theorem states that if p is prime and $\gcd(a, p) = 1$, then $a^{(p-1)} \equiv 1 \pmod{p}$. This theorem provides a necessary condition for primality. Given a candidate integer n, one may select a base a and check whether the corresponding congruence holds. If the congruence fails, n is composite. If it holds, however, the result does not establish that n is prime.

[#]***Corresponding Author***
Key words and phrases*: Fermat pseudoprimes, Fermat's little theorem, Modular arithmetic, computational number theory, Primality testing, Carmichael numbers, Pseudoprime frequency, Modular exponentiation.*

Certain composite integers satisfy the same congruence as primes for one or more testing bases. Such integers are known as Fermat pseudoprimes. Their existence demonstrates that satisfying Fermat's congruence is not a sufficient condition for primality. A classical example is 341 = 11 × 31, which is a Fermat pseudoprime to base 2, since 2^340 ≡ 1 (mod 341). The study of pseudoprimes has a long history in number theory and computational mathematics. Carmichael's work on composite integers satisfying Fermat-type congruences established the importance of distinguishing between base-specific pseudoprimes and numbers that pass the Fermat test for every base relatively prime to them. Later, Alford, Granville, and Pomerance (1994) proved that infinitely many Carmichael numbers exist, demonstrating that the obstruction to Fermat-based primality testing is not limited to a finite collection of exceptional integers.

The computational frequency of pseudoprimes is an important subject in its own right. The number of pseudoprimes below a specified bound measures their cumulative occurrence, while the proportion of pseudoprimes among candidate integers provides a more meaningful basis for comparing different search ranges. Publicly available computational tables document the growth of base-2 Fermat pseudoprimes over ranges extending to at least 10^15 (Jacobsen, 2020). Recent work also continues to examine the asymptotic distribution of Fermat pseudoprimes; Filipovski (2026), for example, revisits the asymptotic density of Fermat pseudoprimes to a fixed base. The present study develops a computational framework for examining these phenomena. It combines mathematical characterization, algorithmic detection, cumulative frequency measurements, and comparisons across numerical bounds. The central objective is not merely to list pseudoprimes, but to investigate how their observed frequency changes as the search range expands and how these observations relate to established results in number theory.

2. Literature Review

Fermat's little theorem is one of the basic results underlying elementary primality testing. For a prime p and an integer a relatively prime to p, it establishes that a^(p−1) ≡ 1 (mod p). The converse does not hold in general, because composite integers may satisfy the same congruence. For example, 341 = 11 × 31 satisfies 2^340 ≡ 1 (mod 341), despite being composite. Consequently, a base-2 Fermat test accepts 341 as a probable prime. This illustrates the distinction between a necessary condition for primality and a sufficient condition. Crandall and Pomerance (2005) provide a systematic treatment of primality testing, pseudoprimes, and computational methods involving modular arithmetic. Carmichael numbers are especially important because they pass Fermat's primality test for every base relatively prime to the number. A classical example is 561 = 3 × 11 × 17. Unlike an ordinary Fermat pseudoprime, which may pass the test for only selected bases, a Carmichael number satisfies the Fermat congruence for every admissible base.

Korselt's criterion characterizes Carmichael numbers: a composite integer n is a Carmichael number if and only if it is square-free and p−1 divides n−1 for every prime divisor p of n. Alford et al. (1994) established that infinitely many Carmichael numbers exist. This result shows that testing multiple bases does not eliminate every

possible Fermat-test false positive. Computational enumeration provides a practical means of studying pseudoprime occurrence over finite ranges. Jacobsen (2020) compiled pseudoprime statistics and tables for several primality tests, including the Fermat test to base 2. The reported cumulative counts provide useful reference values for evaluating computational implementations. The base-2 Fermat pseudoprimes are also catalogued in the On-Line Encyclopedia of Integer Sequences (OEIS), particularly sequence A001567. Related sequences record pseudoprimes according to other properties (OEIS Foundation Inc., n.d.-a, n.d.-b). These resources support independent verification and help identify discrepancies between implementations.
Strong pseudoprimes form a different class: every strong pseudoprime to a given base is a Fermat pseudoprime to that base, but the converse does not hold. This distinction is relevant when evaluating the relative effectiveness of different primality-testing procedures (Jacobsen, 2020). The frequency of pseudoprimes has both computational and probabilistic significance. A useful distinction must be made between the cumulative number of pseudoprimes below a bound and the probability that a randomly selected composite integer passes a particular test. The latter depends on the distribution of candidate integers, the selection of the testing base, and the exact test employed. (Fulara et al., 2025) provide better encryption and decryption capabilities, resilience of keys, improvement of algorithms, and execution in Python, and a profound security analysis and investigation. This research presents a comparative analysis that determines how systems enhanced with pseudoprimes offer distinct advantages over traditional encryption methods. The statistical analysis of pseudoprimes within a given interval is carried out and an algorithm of Python program to find the distribution of pseudoprimes has also been generated, which is used to find their distribution with different bases within the given intervals (Fulara et al., 2024).

Damgård, Landrock, and Pomerance (1993) examine average-case error estimates for the strong probable-prime test. For a fixed base, Fermat pseudoprimes have asymptotic density zero. Filipovski (2026) revisits the asymptotic density of Fermat pseudoprimes to a fixed base. Density zero does not mean that only finitely many pseudoprimes exist; it means that their proportion among integers up to a growing bound tends to zero. Let $a \geq 2$ be an integer and let $n > 1$ be an integer satisfying $\gcd(a, n) = 1$. A composite integer n is a Fermat pseudoprime to base a if $a^{(n-1)} \equiv 1 \pmod{n}$. Wagstaff (2024) investigated relationships between pseudoprimes and Fermat numbers, establishing results about their pseudoprime bases and the structure of the associated sets of bases. It also examined conditions under which strong pseudoprime bases form subgroups of the corresponding groups of Fermat pseudoprime bases. Shallue and Webster (2025) reported an extensive computational enumeration of Carmichael numbers, identifying 49,679,870 such numbers below 10^{22}. Their algorithm improved substantially upon earlier computational tabulations by exploiting the multiplicative structure of Carmichael numbers of the form $n=pqr$. It also highlights the value of computational efficiency, independent verification, and reproducible enumeration in research on pseudoprime distributions. Li and Pomerance (2026) established an upper bound for the distribution of base-a Fermat pseudoprimes that is uniform in the testing base and

does not require the usual co-primality restriction in the formulation of their result. They also proved the existence of infinitely many near-Carmichael numbers, which behave as pseudoprimes for a positive proportion of bases but not for all bases.

Agarwal et al. (2026) proposed an algorithmic framework for generating and classifying pseudoprimes within a finite search interval. The study considers Fermat, Euler–Jacobi, and strong pseudoprimes and discusses the special role of Carmichael numbers. Its emphasis on classification and computational procedures is relevant to the reproducibility of pseudoprime-frequency research. Alcantarilla Sánchez et al. (2026) investigated Euler pseudoprimes derived from Carmichael numbers and developed an algorithm to identify composite integers that survive the Solovay–Strassen primality test for numerous bases. Boltnev (2025) studied the construction of Carmichael numbers that also behave as strong pseudoprimes for multiple initial prime bases in the Miller–Rabin test. This research addresses a more restrictive class of composite numbers than ordinary Fermat pseudoprimes. Luca et al. (2025) investigated pseudoprime RSA moduli and their relationship with factorization in public-key cryptography. Their study illustrates how pseudoprime-related properties can intersect with the arithmetic structure of composite integers used in cryptographic systems. Although RSA moduli and base-specific Fermat pseudoprimes represent different mathematical objects, both are relevant to computational number theory.

Pomerance (1981) derived quantitative upper bounds for the distribution of pseudoprimes to base 2. The results provide a theoretical framework for estimating how the number of pseudoprimes grows relative to a numerical search bound. Grantham (2001) developed the theory of Frobenius pseudoprimes and examined their role in primality testing. The study represents an important extension of pseudoprime research beyond the elementary congruence associated with Fermat's little theorem. Damgård et al. (1993) investigated average-case error estimates for the strong probable-prime test. Their research is relevant to the assessment of primality-testing reliability because it focuses on the probability of incorrectly accepting a composite integer under a stronger testing procedure. The findings provide a theoretical point of comparison for research on Fermat pseudoprimes, where the primary concern is the occurrence of composites satisfying a particular congruence. Rabin (1980) developed a probabilistic algorithm for primality testing, contributing to the development of efficient methods for identifying probable primes. This research provides a useful comparison with elementary Fermat testing, which can incorrectly accept composite integers that satisfy the relevant congruence.

The problem addressed in this paper is to determine the number of such integers below a given bound x, investigate the corresponding normalized frequency, and examine how these quantities vary with the testing base and search range.

3. Research Gap

Existing theoretical work establishes important properties of Fermat pseudoprimes, while computational tables provide extensive information about their occurrence. A useful computational investigation should bring these perspectives together by specifying the counting criterion, explaining frequency normalization, comparing

numerical ranges, and providing reproducible procedures. The present paper integrates mathematical definitions, cumulative statistics, normalized frequency measures, and computational procedures in a single study.

4. Research Methodology

Research Design

The study follows a computational number-theory approach, combining mathematical analysis with numerical enumeration and interpretation of published reference data. The investigation has three stages:

- Mathematical specification of the definition and counting function
- Computational enumeration using modular exponentiation and candidate filtering
- Statistical interpretation through cumulative counts and normalized frequencies

Published pseudoprime counts are treated as reference benchmarks rather than as results independently generated by the present author. A full empirical replication requires execution of the supplied algorithm, recording the software environment, and validating the resulting counts.

Choice of Bases

The principal investigation considers base 2, a common reference base in computational pseudoprime tables. The procedure can be repeated for bases 3, 5, 7, and others. When comparing bases, the eligibility condition gcd(a, n) = 1 must be enforced. A candidate with a nontrivial common divisor is composite, but it is not counted as a Fermat pseudoprime to that base under the adopted definition.

Computational Criterion

Step-1: Determine whether n is composite.

Step-2: Calculate g = gcd(a, n).

Step-3: Exclude n if g is not equal to 1.

Step-4: Compute r = a^(n−1) mod n using modular exponentiation.

Step-5: Count n as a pseudoprime if and only if r = 1.

Modular exponentiation is essential because directly calculating a^(n−1) becomes infeasible for large n. Repeated squaring computes the modular residue efficiently without constructing the full power.

Frequency Measures

- $P_a(x)$: cumulative number of Fermat pseudoprimes.
- $F_a(x) = P_a(x)/x$: pseudoprimes per integer in the complete interval.

- $Q_a(x) = P_a(x)/C_a(x)$: pseudoprimes as a proportion of eligible composite candidates.
- $P_a(x) - P_a(y)$: number of pseudoprimes in (y, x], for $y < x$.

These measures distinguish cumulative growth from local occurrence. Because $P_a(x)$ is non-decreasing, its absolute value cannot decrease as the upper bound grows. Normalized measures need not be monotonic at every finite bound.

Data Sources and Verification

The principal external benchmark is Dana Jacobsen's published table, Pseudoprime Statistics and Tables, which provides cumulative counts for base-2 Fermat pseudoprimes and describes the computational provenance of the data (Jacobsen, 2020). OEIS sequence A001567 provides a further reference for base-2 Fermat pseudoprimes (OEIS Foundation Inc., n.d.-a). These resources support cross-checking, but agreement with a published table does not replace a record of the author's own computation. For reproducibility, retain the programming language and version, operating system and processor, exact upper bounds and bases, composite-number detection method, modular exponentiation implementation, runtime and memory consumption, output counts, and independent validation procedure.

5. Computational Algorithm

Algorithmic Description

Enumeration of Fermat pseudoprimes.

Input: base $a \geq 2$ and upper bound X.

Output: list of Fermat pseudoprimes to base a not exceeding X.

- Initialize an empty list L.
- For each integer n from 2 to X, determine whether n is prime; if prime, continue.
- If gcd(a, n) is not equal to 1, continue.
- Compute r = pow(a, n−1, n), where the third argument specifies modular exponentiation.
- If r = 1, append n to L.
- Return L and its cardinality |L|.

The primality check must be exact over the tested range. A deterministic sieve or a rigorously deterministic primality test for the relevant integer range can be used. A probable-prime test should not be the sole basis for claiming that every rejected candidate is prime.

Python Implementation

```
from math import gcd, isqrt

def is_prime(n):
    if n < 2:
        return False
    if n in (2, 3):
        return True
    if n % 2 == 0:
        return False

    for d in range(3, isqrt(n) + 1, 2):
        if n % d == 0:
            return False
    return True

def fermat_pseudoprimes(limit, base=2):
    result = []
    for n in range(2, limit + 1):
        if is_prime(n):
            continue
        if gcd(base, n) != 1:
            continue
        if pow(base, n - 1, n) == 1:
            result.append(n)
    return result

if __name__ == "__main__":
    for base in (2, 3, 5, 7):
        values = fermat_pseudoprimes(100_000, base)
        print(f"Base: {base}, limit: 100000, count: {len(values)}")
```

Python's three-argument pow performs modular exponentiation efficiently. Nevertheless, the primality routine above uses trial division and becomes slow as the search limit increases. Larger studies should use a segmented sieve, optimized exact primality routines, or established computational number-theory libraries, with appropriate verification of their output.

Validation of algorithm

For base 2, 341 is a useful test case. Since 341 = 11 × 31, it is composite. Moreover, 2^10 ≡ 1 (mod 341), and 340 = 34 × 10, so 2^340 ≡ 1 (mod 341). Therefore, a correct implementation must identify 341 as a Fermat pseudoprime to base 2.

A second validation case is 561 = 3 × 11 × 17. This is a Carmichael number and must be accepted by the Fermat test for every base relatively prime to 561. These

examples provide basic checks of the congruence and base-selection logic, although they are not sufficient by themselves to validate a large-scale enumeration.

6. Computational Results and Analysis

The published pseudoprime statistics compiled by Jacobsen (2020) report the following cumulative counts for Fermat pseudoprimes to base 2.

Upper bound x	$P_2(x)$	Count per 10^9 integers
10^9	5,597	5,597
10^{10}	14,884	1,488.4
10^{11}	38,975	389.75
10^{12}	101,629	101.629
10^{13}	264,239	26.4239
10^{14}	687,007	6.87007
10^{15}	1,801,533	1.801533

Source: Jacobsen (2020), "Pseudoprime Statistics and Tables," base-2 Fermat pseudoprime column. The final column is calculated by dividing each cumulative count by x and multiplying by 10^9.

These figures are published computational benchmarks, not newly generated experimental results of this paper. They provide a useful quantitative basis for discussing pseudoprime frequency.

Growth in Absolute Counts

The table shows that the cumulative number of base-2 Fermat pseudoprimes increases as the search bound expands. The count rises from 5,597 below 10^9 to 101,629 below 10^12, and reaches 1,801,533 below 10^15. This behavior is consistent with $P_2(x)$ as a cumulative counting function: every pseudoprime already counted below a bound remains included when the bound is increased. The increase in the number of pseudoprimes does not imply that pseudoprimes become more common relative to all integers. Absolute count and relative frequency are different quantities, and both are needed to interpret the numerical results correctly.

Normalized Frequency

The final column expresses the cumulative count as pseudoprimes per billion integers in the interval up to x. This normalization reveals a marked decline as the search bound increases. There are approximately 5,597 base-2 Fermat pseudoprimes per billion integers up to 10^9, compared with approximately 1.80 per billion integers up to 10^15. These are cumulative proportions, not counts of pseudoprimes within the final billion integers of each range. The observed decline illustrates why reporting only absolute counts can be misleading. Although the number of pseudoprimes grows substantially, the total number of integers grows much faster in these benchmark ranges. The result is consistent with the established density-zero property of Fermat pseudoprimes to a fixed base. A finite table, however, cannot prove that asymptotic property.

Increase across Successive Ranges

Interval	Additional pseudoprimes
$(10^9, 10^{10}]$	9,287
$(10^{10}, 10^{11}]$	24,091
$(10^{11}, 10^{12}]$	62,654
$(10^{12}, 10^{13}]$	162,610
$(10^{13}, 10^{14}]$	422,768
$(10^{14}, 10^{15}]$	1,114,526

Source: Calculated from the cumulative benchmark counts in Jacobsen (2020).

The interval counts increase because each successive interval spans a much larger numerical range. They should not be compared without accounting for interval length. A rigorous local-frequency comparison would divide each interval count by the number of eligible composite candidates in that interval.

Interpretation of the Frequency Pattern

Three conclusions can be drawn from the benchmark data. First, Fermat pseudoprimes are sufficiently numerous to be relevant to computational primality testing. Second, the absolute number and relative proportion exhibit different trends: the cumulative count increases, whereas normalized cumulative frequency decreases over the reported bounds. Third, these patterns are base-specific; the reported values apply to base 2 and cannot be transferred directly to another base without further computation. The results underline the importance of distinguishing observed frequency from asymptotic density. Numerical data illustrate the distribution over finite ranges, while theoretical results address behavior as the bound tends to infinity.

7. Comparative Analysis of Testing Bases

Dependence on the Testing Base

For a fixed bound x, the count P_a(x) depends on the chosen base a. Two different bases generally define different sets of pseudoprimes. A comparative study should calculate each counting function separately and report raw counts and normalized proportions for common upper bounds.

Multiple Bases Improve Detection

If a composite integer passes the Fermat test to base 2, testing it with another base can reject it if the corresponding congruence fails. However, testing several bases does not guarantee primality. Carmichael numbers pass the Fermat test for every base co-prime to them, and infinitely many Carmichael numbers exist (Alford et al., 1994). This limitation should be distinguished from the behavior of stronger tests. The Miller–Rabin test uses additional congruence conditions and has a different set of pseudoprimes. Every strong pseudoprime to a given base is also a Fermat pseudoprime to that base, but the converse does not hold (Jacobsen, 2020).

Comparative Experiment

Quantity	Purpose
$P_a(x)$	Measures the absolute number of pseudoprimes
$P_a(x)/x$	Measures frequency among all integers
$P_a(x)/C_a(x)$	Measures frequency among eligible composite candidates
$P_a(x) - P_a(y)$	Measures occurrence in a specified interval
Runtime	Measures computational cost
Intersection of base-specific sets	Measures overlap between pseudoprime sets

For a fair comparison, all bases should use the same upper bounds, the same definition of a pseudoprime, and comparable implementations. The computing environment should be reported when runtime is used as a performance measure.

8. Discussion

The computational evidence demonstrates that the frequency of Fermat pseudoprimes must be interpreted through more than one statistical measure. Cumulative counts increase with the search bound, while normalized cumulative proportions in the published base-2 table decline substantially. This pattern is consistent with the known asymptotic density-zero result for pseudoprimes to a fixed base. The distinction between density and infinitude is particularly important. A set can contain infinitely many elements and nevertheless have density zero. Therefore, declining normalized frequencies do not imply that pseudoprimes eventually disappear. Rather, they indicate that pseudoprimes become progressively less common relative to all positive integers. The results also illustrate a practical limitation of Fermat-based primality testing. The Fermat congruence is computationally convenient, but it cannot by itself distinguish every composite integer from a prime. Pseudoprimes provide explicit counterexamples, and Carmichael numbers demonstrate that the difficulty persists even when several bases are tested. From an algorithmic perspective, modular exponentiation makes the congruence test efficient for individual candidates. An exhaustive search over an enormous interval also requires an efficient method for identifying composite numbers and managing the search. A naive implementation based on trial division is appropriate for demonstration and small-scale verification but is not suitable for reproducing the largest published tables.

Published computational data can support a literature-based numerical analysis, but they should not be described as the outcome of a new computation unless the calculation has actually been performed. This distinction matters for scientific reproducibility and for the validity of claims concerning computational performance. The framework can be extended by computing local frequencies over intervals of equal length or geometrically increasing intervals, comparing pseudoprime frequencies across bases, and comparing Fermat pseudoprimes with strong pseudoprimes and Carmichael numbers. Finite numerical experiments can reveal patterns and suggest conjectures, but universal claims about distribution require mathematical proof.

9. Conclusion

This paper has presented a mathematical and computational framework for investigating the frequency of Fermat pseudoprimes. By defining a base-specific counting function, distinguishing absolute counts from normalized frequencies, and presenting a reproducible enumeration procedure, the study clarifies the principal methodological requirements for examining pseudoprime distribution. Published computational benchmarks for base-2 Fermat pseudoprimes show that the cumulative count increases from 5,597 below 10^9 to 1,801,533 below 10^15. At the same time, normalized cumulative frequency decreases substantially across these ranges. The observations demonstrate the importance of reporting both absolute counts and relative frequencies.

The analysis also highlights the limitations of Fermat's primality test. Composite integers can satisfy the Fermat congruence for selected bases, while Carmichael numbers satisfy it for every base coprime to the number. The existence of infinitely many Carmichael numbers establishes that these exceptions are a fundamental feature of the test rather than a finite collection of anomalies. The computational framework presented here provides a basis for independent verification and further investigation. To establish a fully original empirical contribution, future work should execute the algorithm over explicitly specified ranges, document the computational environment, report independently verified counts, and compare local and cumulative frequencies across several bases. Such extensions would strengthen the connection between computational evidence and the theoretical study of pseudoprimes.

References

1. Alford, W. R., Granville, A., & Pomerance, C. (1994). There are infinitely many Carmichael numbers. Annals of Mathematics, 139(3), 703–722. https://doi.org/10.2307/2118576
2. Crandall, R., & Pomerance, C. (2005). Prime numbers: A computational perspective (2nd ed.). Springer. https://doi.org/10.1007/978-1-4684-9316-0
3. Damgård, I., Landrock, P., & Pomerance, C. (1993). Average case error estimates for the strong probable prime test. Mathematics of Computation, 61(203), 177–194.
4. Filipovski, S. (2026). On the asymptotic density of Fermat pseudoprimes to a fixed base. The American Mathematical Monthly. Published online September 16, 2026. https://doi.org/10.1080/00029890.2026.2722582
5. Fulara, B., Bhatt, A., Agarwal, S., Sharma, D.K. (2025). Enhancing Cryptographic Security Using Pseudoprimes: A Novel Encryption-Decryption Algorithm, International Conference on Innovations and Emerging Technologies In AI & Communication Systems (IETACS), pp. 63-66, IEEE. https://doi.org/10.1109/IETACS68750.2025.11385543
6. Jacobsen, D. (2020, March 31). Pseudoprime statistics and tables. ntheory.org. https://ntheory.org/pseudoprimes.html
7. OEIS Foundation Inc. (n.d.-a). Sequence A001567: Fermat pseudoprimes to base 2. The On-Line Encyclopedia of Integer Sequences. https://oeis.org/A001567

8. OEIS Foundation Inc. (n.d.-b). Sequence A086249: Number of base-2 Fermat pseudoprimes having a specified multiplicative order. The On-Line Encyclopedia of Integer Sequences. https://oeis.org/A086249
9. Fulara, B., Bhatt, A., Sharma, D. K., Agarwal, S., Mathpal, G., & Mathpal, R. (2024). Statistical Analysis and Distribution of Fermat Pseudoprimes Within the Given Interval. International Journal of Experimental Research and Review, 44, 115–120.https://doi.org/10.52756/ijerr.2024.v44spl.010
10. Wagstaff, S. S., Jr. (2024). Pseudoprimes and Fermat numbers. INTEGERS, 24, Article A23. https://doi.org/10.5281/zenodo.10680422
11. Shallue, A., & Webster, J. (2025). Advances in tabulating Carmichael numbers. Research in Number Theory, 11, Article 8. https://doi.org/10.1007/s40993-024-00598-3
12. Li, S., & Pomerance, C. (2026). Fermat pseudoprimes. Acta Arithmetica, 222, 191–196. https://doi.org/10.4064/aa250411-9-10
13. Agarwal, S., Singh, S. K., Sharma, D. K., Fulara, B., Jangid, V., Sharma, A., & Dwivedi, S. (2026). An algorithmic approach to generating and classifying pseudoprimes. Journal of Business Management and Information Systems, 13(1), 45–54. https://doi.org/10.48001/jbmis.1301004
14. Alcantarilla Sánchez, A., Cottaar, J., Lange, T., & de Weger, B. (2026). Ours go to 211: Euler pseudoprimes to 47 prime bases (from Carmichael numbers). arXiv preprint arXiv:2602.21840. https://doi.org/10.48550/arXiv.2602.21840
15. Boltnev, Y. F. (2025). Construction of strong pseudoprime Carmichael numbers passing the Miller–Rabin test for many first prime bases. Prikladnaya Diskretnaya Matematika. Supplement, 18, 258–262. https://doi.org/10.17223/2226308X/18/55
16. Luca, F., Ralaivaosaona, D., & Jiménez Urroz, J. (2025). On pseudoprime RSA moduli. Afrika Matematika, 36, Article 118. https://doi.org/10.1007/s13370-025-01338-1
17. Pomerance, C. (1981). On the distribution of pseudoprimes. Mathematics of Computation, 37(156), 587–593. https://doi.org/10.1090/S0025-5718-1981-0628717-0
18. Grantham, J. (2001). Frobenius pseudoprimes. Mathematics of Computation, 70(234), 873–891. https://doi.org/10.1090/S0025-5718-00-01197-2
19. Damgård, I., Landrock, P., & Pomerance, C. (1993). Average case error estimates for the strong probable prime test. Mathematics of Computation, 61(203), 177–194.
20. Rabin, M. O. (1980). Probabilistic algorithm for testing primality. Journal of Number Theory, 12(1), 128–138. https://doi.org/10.1016/0022-314X(80)90084-0

[1#]**SHUBHAM AGARWAL**: PROFESSOR, DEPARTMENT OF MATHEMATICS, NEW DELHI INSTITUTE OF MANAGEMENT, NEW DELHI, INDIA, (EMAIL: meshubhamagarwal@gmail.com)

[2]**BHAWANA FULARA**: RESEARCH SCHOLAR, DEPARTMENT OF MATHEMATICS, UTTARAKHAND OPEN UNIVERSITY, HALDWANI, INDIA, (EMAIL: bhawana.fulara83@gmail.com)

[3]**DEEPAK KUMAR SHARMA**: ASSISTANT PROFESSOR, DEPARTMENT OF MATHEMATICS, UTTARAKHAND OPEN UNIVERSITY, HALDWANI, INDIA, (EMAIL: sharmadeepak0111209@gmail.com)

[4]**ARVIND BHATT**: PROFESSOR, DEPARTMENT OF MATHEMATICS, UTTARAKHAND OPEN UNIVERSITY, HALDWANI, INDIA, (EMAIL: arvindbhatt@uou.ac.in)