NUMBER THEORY

Modular Arithmetic

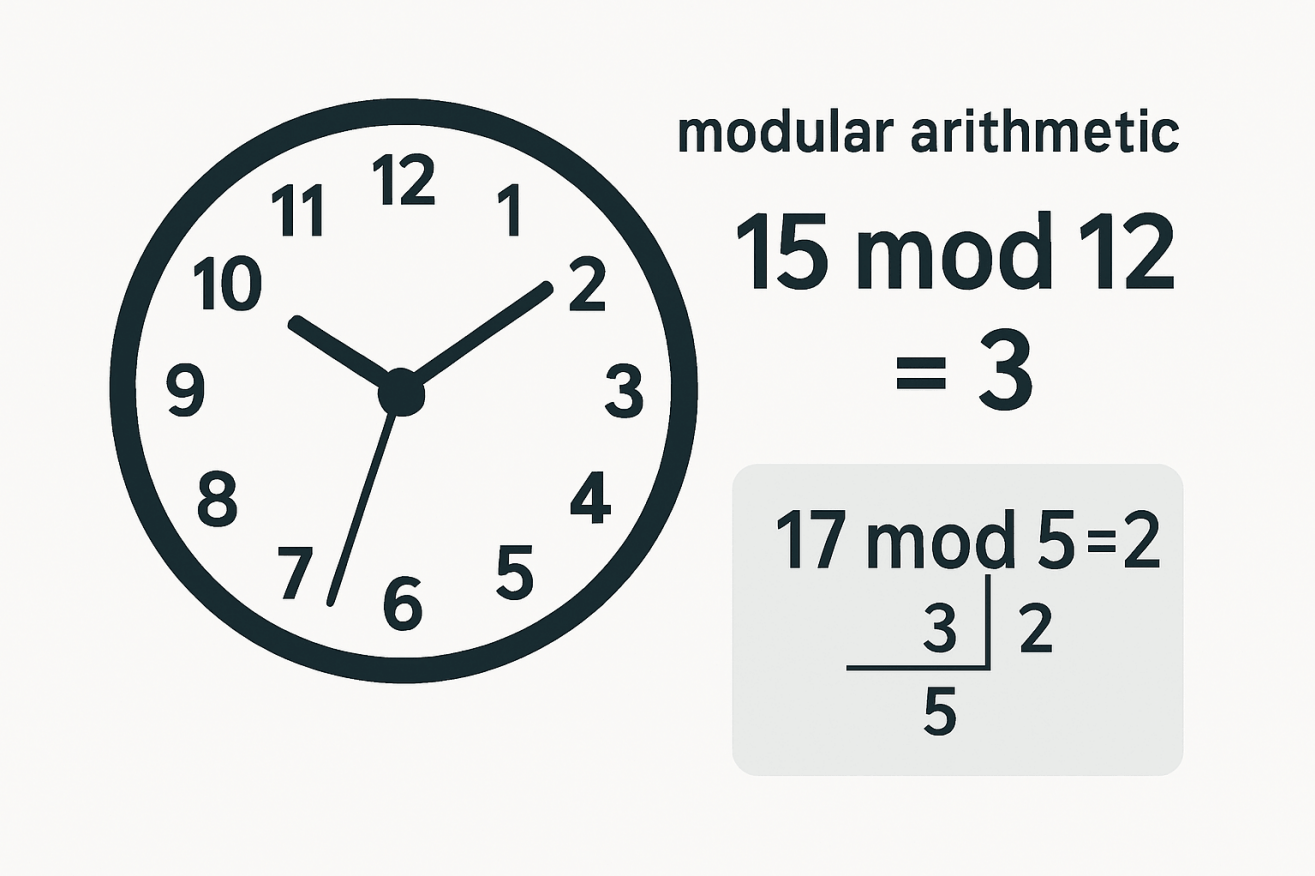
Clock arithmetic made precise: congruences, remainders, divisibility tests, and a 10-question practice set with full solutions.

SUBJECT	LEVEL	INCLUDES
Mathematics	High school and early college	Practice set, answer key, revision sheet

Read this note online, with updates: gauravtiwari.org/study-notes/modular-arithmetic

INSIDE

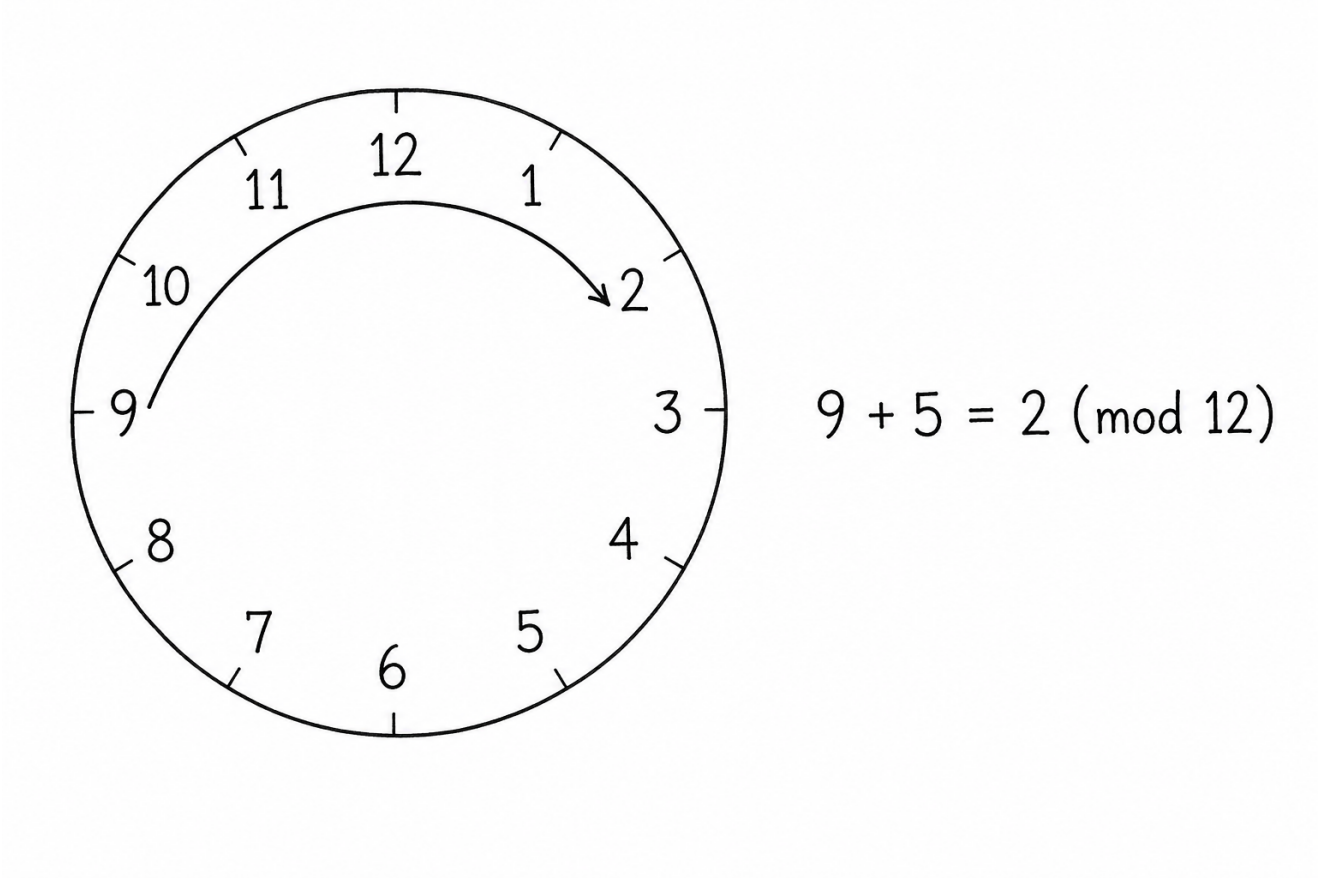
Modular arithmetic, sometimes called 'clock arithmetic', is a system of arithmetic for integers in which numbers wrap around after reaching a fixed value called the modulus. If you take 15 hours past midnight on a 12-hour clock, you don't land at 15 o'clock, you land at 3 o'clock, because clocks wrap around every 12 hours. That's modular arithmetic. Formalized by Gauss in Disquisitiones Arithmeticae (1801), it's the language of cryptography, hash functions, digital signal processing, and much of computer science.



Modular arithmetic wraps numbers around a modulus, like hours on a clock face.

Definition

Two integers a and b are **congruent modulo** n (written $a \equiv b \pmod{n}$) if their difference is divisible by n. In symbols:



Clock arithmetic: 5 hours after 9 o'clock is 2 o'clock, because 14 and 2 are the same thing mod 12.

$$a \equiv b \pmod{n} \iff n \mid (a - b)$$

Equivalently, a and b leave the same remainder when divided by n. The number n is the *modulus*.

Examples. $17 \equiv 5 \pmod{12}$ because $17 - 5 = 12$ is divisible by 12. $38 \equiv 3 \pmod{7}$ because $38 = 5 \cdot 7 + 3$. $-1 \equiv 11 \pmod{12}$ because $-1 - 11 = -12$ is divisible by 12.

Operations Mod n

Addition, subtraction, and multiplication all work normally modulo n:

- $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$.
- $(a - b) \bmod n$, same rule.
- $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$.
- **Division is trickier:** you can only divide by b modulo n when $\gcd(b, n) = 1$, and division is done by finding the *modular inverse* of b.

Worked Examples

Day-of-week calculation. Today is Wednesday. What day is it 100 days from now? Take $100 \bmod 7 = 2$. Add 2 days to Wednesday: Friday.

Last digit. The last digit of 7^{100} is $7^{100} \bmod 10$. Powers of 7 mod 10 cycle: $7, 9, 3, 1, 7, 9, 3, 1, \ldots$ with period 4. $100 \bmod 4 = 0$, so $7^{100} \equiv 7^4 \equiv 1 \pmod{10}$. Last digit is 1.

Solving a congruence. Solve $3x \equiv 7 \pmod{11}$. The inverse of 3 mod 11 is 4 (because $3 \cdot 4 = 12 \equiv 1 \pmod{11}$). So $x \equiv 4 \cdot 7 = 28 \equiv 6 \pmod{11}$.

Fermat's Little Theorem

If p is prime and a is not divisible by p:

$$a^{p-1} \equiv 1 \pmod{p}$$

Equivalently, $a^p \equiv a \pmod{p}$ for all integers a (including those divisible by p). This is a workhorse for computing very large powers modulo a prime and is the basis of many primality tests.

Example. Compute $2^{100} \bmod 13$. By Fermat's little theorem, $2^{12} \equiv 1 \pmod{13}$. And $100 = 8 \cdot 12 + 4$, so $2^{100} = (2^{12})^8 \cdot 2^4 \equiv 1 \cdot 16 \equiv 3 \pmod{13}$.

Applications

- **RSA encryption.** RSA computes ciphertext as message^e mod n, where n = pq is the product of two large primes and e is the public exponent. Decryption uses message^d mod n, where d is the private exponent. Everything happens in modular arithmetic.
- **Hash functions.** Hash tables use h(key) mod table_size to map keys to bucket indices. Prime table sizes minimize clustering.
- **Check digits.** ISBN-10 check digit uses arithmetic mod 11; credit card numbers (Luhn algorithm) use

a mod 10 check.

- **Pseudo-random number generators.** Linear congruential generators (LCGs) compute $X_{n+1} = (a \cdot X_n + c) \bmod m$. Choice of a, c, m determines the quality of the output sequence.
- **Cyclic structures.** Days of the week, hours, calendars, angles (mod 360°), musical pitches (mod 12), anything that repeats periodically is naturally described with modular arithmetic.
- **Error-correcting codes.** Reed-Solomon and CRC checksums operate in modular arithmetic (often over finite fields with prime modulus).

Related study notes: Prime Numbers, Euclidean Algorithm, Cryptography, Fundamental Theorem of Algebra.

Practice Questions

Work each question before reading its solution. The set runs from direct recall and substitution to the applied questions that exams actually use to separate grades.

Q1. Compute $17 \bmod 5$, $100 \bmod 7$, and $-3 \bmod 8$.

SOLUTION

$17 = 3 \times 5 + 2$, so 2. $100 = 14 \times 7 + 2$, so 2. For -3: add 8 to reach the standard range, giving 5, since $-3 = -1 \times 8 + 5$. The result always lands in 0 to $m - 1$, negatives included.

Q2. It is 9 o'clock. What time does a 5-hour meeting end, and what computation is that?

SOLUTION

$9 + 5 = 14 \equiv 2 \pmod{12}$: it ends at 2 o'clock. Clock time is arithmetic modulo 12; days of the week are modulo 7. You have used modular arithmetic your whole life without the notation.

Q3. What does $a \equiv b \pmod{m}$ mean? Give the 2 equivalent formulations.

SOLUTION

Either: a and b leave the same remainder on division by m. Or: m divides $a - b$. The second is usually easier to use in proofs: $37 \equiv 2 \pmod{5}$ because $5 \mid 35$.

Q4. Find the last digit of 7^{100}.

SOLUTION

Last digits are mod 10. Powers of 7 cycle: $7, 9, 3, 1, 7, 9, \ldots$ with period 4. Since $100 = 4 \times 25$, the exponent lands at the cycle's end: last digit 1. Huge powers collapse to short cycles: that is the whole trick.

Q5. Compute $3^{50} \bmod 7$ without ever holding a large number.

SOLUTION

Reduce as you go. $3^6 \equiv 1 \pmod{7}$ (Fermat's little theorem, or check: $3^2 = 2$, $3^3 = 6$, $3^6 = 6^2 = 36 \equiv 1$). Then $3^{50} = 3^{48} \cdot 3^2 = (3^6)^8 \cdot 9 \equiv 1 \cdot 2 = 2 \pmod{7}$.

Q6. Prove the divisibility-by-9 test: a number is divisible by 9 exactly when its digit sum is.

SOLUTION

Since $10 \equiv 1 \pmod 9$, every power $10^k \equiv 1$. A number $d_n \cdots d_1 d_0 = \sum d_k 10^k \equiv \sum d_k \pmod 9$: the number and its digit sum are congruent, so one is divisible by 9 exactly when the other is. The digit test is modular arithmetic in costume.

Q7. Solve $x + 7 \equiv 3 \pmod{10}$ and then $3x \equiv 4 \pmod 7$.

SOLUTION

First: $x \equiv 3 - 7 = -4 \equiv 6 \pmod{10}$. Second: multiply by the inverse of 3 mod 7, which is 5 (because $15 \equiv 1$): $x \equiv 5 \times 4 = 20 \equiv 6 \pmod 7$. Division works only through inverses, and an inverse exists only when gcd of the coefficient and modulus is 1.

Q8. Why does an ISBN-10 check digit use arithmetic mod 11, and what does it catch?

SOLUTION

The check digit makes $\sum_{i=1}^{10} i \cdot d_i \equiv 0 \pmod{11}$. Because 11 is prime, every single-digit error and every adjacent-digit swap changes the weighted sum by a nonzero amount mod 11, so both error types are always detected. Composite moduli would let some errors slip through.

Q9. Compute $2^{10} \bmod 1000$, then explain what "the last 3 digits of 2^n" means modularly.

SOLUTION

$2^{10} = 1024 \equiv 24 \pmod{1000}$. The last 3 digits of any number are exactly its residue mod 1000, just as the last digit is its residue mod 10. Digit questions are always modulus questions.

Q10. Where does modular arithmetic run in your daily life? Name 3 systems.

SOLUTION

RSA encryption performs its encryption and decryption as modular exponentiation with 600-digit moduli, securing effectively every HTTPS connection. Hash tables place keys in buckets by hashing mod the table size. Check digits on ISBNs, credit cards (Luhn, mod 10), and bank codes catch typos. Clocks, calendars, and cyclic schedules round out the list.

Answer Key

QUICK ANSWERS

Q1 2, 2, 5 **Q2** 2 o'clock: $14 \equiv 2 \pmod{12}$ **Q3** same remainder; $m \mid (a-b)$ **Q4** 1 **Q5** 2

Q6 $10^k \equiv 1 \pmod 9$ makes number ≡ digit sum **Q7** $x \equiv 6 \pmod{10}$; $x \equiv 6 \pmod 7$ **Q8** prime modulus catches all single errors and swaps **Q9** 24 **Q10** cryptography, hash tables, check digits

Revision Sheet

The congruence

$a \equiv b \pmod m$ iff $m \mid (a-b)$ iff same remainder. Results live in $0..m-1$.

Arithmetic rules

Add, subtract, multiply freely, reducing at any time. Divide only via modular inverses (need gcd = 1).

Big powers

Find the cycle of powers, or use Fermat: $a^{p-1} \equiv 1 \pmod p$ for prime p, $p \nmid a$.

Digit tests

Mod 9 and 3: digit sum (since $10 \equiv 1$). Mod 11: alternating sum. Last k digits: mod 10^k.

Negatives

Add the modulus until in range: $-3 \equiv 5 \pmod 8$.

Where it runs

RSA/HTTPS, hash tables, check digits, clocks and calendars.

About These Notes

This note is part of a free study-notes series on gauravtiwari.org covering mathematics, physics, chemistry, and biology: the concepts that keep deciding grades in high school, board, and entrance exams.

2008	2,000+	125+	M.Sc.
Writing and teaching online since	Articles published on gauravtiwari.org	Free study notes in this series	Mathematics, with physics and chemistry training

Gaurav Tiwari is a developer, educator, and founder of Gatilab. The study-notes series exists because most exam prep material is either a full textbook or a thin definition page, and revision needs something in between: one concept, complete, with the traps named and the practice attached.

GET THE FULL SERIES

Every note in the series is free to read online and free to download as a PDF like this one. Browse the complete collection at **gauravtiwari.org/free-study-notes-exam-prep** and pick the topics your next exam actually covers.

USE IT FREELY

This PDF is free for personal study, classrooms, and study groups. Share the file or the link with anyone it can help. The one thing not allowed is reselling it or republishing it as your own work.

Gaurav Tiwari

gauravtiwari.org