AMC JOURNAL (DHANGADHI)
Year 9; July, 2026; Vol. 8; Issue 2; PP. (1-12) DOI: https://doi.org/10.3126/amcjd.v8i2.98370

Congruence Modulo *n*: Algebraic Foundations and Consequences for Error Detection

Gaj Ram Damai
Department of Mathematics
Siddhanath Science Campus, Mahendranagar, Tribhuvan University, Nepal
Email: gaj.damai@snsc.tu.edu.np
ORCID: 0000-0002-6560-4863.

Article History:
Received: March 5, 2026
Revised: May 12, 2026
Accepted: June 20, 2026

Abstract

This paper first reviews the concept of congruence modulo *n* in number theory, contrasting its modern algebraic formulation with the ancient concept of modular arithmetic. Practical applications include solving linear congruences, clock arithmetic (modulo 7 and 12), and check-digit verification for ISBN (modulo 11) and bank identification numbers (modulo 10). Building on these, we introduce novel generalizations for error detection in identification numbers. We define an *error-detecting weight modulus pair* (*W*, *m*) and prove necessary and sufficient conditions for detecting all single-digit and adjacent transposition errors: *m* must be prime to detect all transpositions, and weight vector entries must be distinct and non-zero modulo *m* to detect single-digit errors. These results enable more robust check-digit systems for banking, ISBN, and digital identity frameworks. Finally, it gives the future line of research.

Key Words: Residue Class modulo n; Check digit; Error Detection; Bank Identification Numbers (Modulo 10).

MSC 2020: 08B10; 97F60

Introduction

The concept of congruence modulo *n* is a fundamental notion in number theory. Its precise idea wasn't formally established until relatively recently in the history of mathematics. The underlying principles of modular arithmetic have been understood for centuries. Modular arithmetic evolved from informal, practical applications in ancient cultures—such as calendar calculations—into a formal mathematical system. A classical and direct application of congruence modulo an ideal is the Chinese Remainder Theorem, was used in early problem-solving contexts to determine solutions to systems of simultaneous congruencies (Shoup, 2012; Burton, 2007; Lang, 2002). Later mathematicians, including Euler, explored these ideas using the language of divisibility. In 1801, Gauss revolutionized the field by introducing the modern congruence notation "≡" and developing a systematic theory in his *Disquisitiones Arithmeticae*. This formalization became a cornerstone of modern number theory and has found extensive applications in science and technology. The set of integers modulo *n* provides a fundamental example in abstract algebra and is essential for investigating various algebraic structures, including groups and rings (Thomas & Rober, 2020). This paper discusses the basic properties of congruence modulo *n* and the associated notion of residue classes modulo *n*. Additional topics include the Chinese Remainder Theorem is a result from elementary number theory about the solution of systems of simultaneous congruencies (Rosen, 2019). The Chinese mathematician Suntsï wrote about the theorem in the first century A.D. This theorem has some interesting consequences in the design of software for parallel processors, Euler's function $\varphi(n)$, Euler's theorem, Fermat's little theorem, quadratic residues, and summations over divisors (Tuset, 2024; Shoup, 2012; Burton, 2007). In the context of ring theory, as

presented in the 1965 edition of Serge's *Algebra*, congruence modulo an ideal serves as the foundation for constructing factor rings. Specifically, congruence modulo an ideal is the basis for the Chinese Remainder Theorem in the setting of commutative rings (Lang, 2002; Gallian, 2021).

Objectives

The objectives of this research paper are:

a) Gather all the basic definitions, rules, properties, example problems, and real-life uses that are related to this topic.

b) Write the error-detection problem (using weighted sums and remainders after division by *m*) in a clear and exact mathematical way.

c) Find the exact rules for choosing the number ***m*** and the weights so that **every single-digit mistake** is always caught.

d) Find the exact rules for choosing m and the weights so that every swap of two neighboring digits is always caught**.**

Motivation and Problem Statement (Harel, 2013; Booth et al, 2016).

For a long time, people have used remainders after division by a number *n* to create check digits for ID numbers—like the ISBN-10, system which uses modulo 11, or bank routing numbers which use modulo 10.

But the systems we have now just pick a fixed number and fixed weights without any solid math reason to prove they are the best choice. This made us ask two main research questions:

a) What exact math rules must the number *m* and the list of weights follow so that every single-digit mistake is always caught?

b) What rules make sure that every swap of two digits that are next to each other is always caught?

Novel Contributions (Swales, 2014).

This paper makes the following original contributions:

a) It gives a clear definition of what we call an "error-detecting weight-modulus pair" (written as **(**W, m) and puts the rules for detecting errors into a proper mathematical framework.

b) It presents new mathematical rules (theorems) that tell us exactly when all single-digit errors are caught, and exactly when all adjacent swaps are caught. It also provides the step-by-step reasoning (proofs) to show why these rules work.

c) It also adds one extra result (a corollary) that follows directly from these main rules.

Materials and Methods

We use following method for new work.

We pick a number m and assign a special number, called a **weight**, to each digit position. For position i, the weight is written as w_i, and the digit is x_i. The check value **C** is the remainder we get after dividing the sum of all weighted digits by *m*:

C $= \sum_{i=1}^{k} w_i x_i$, mod *m* (The check function)

Single-digit errors:

If only one digit at position ***i*** changes from x_i to x'_i, the new check value **C′** becomes:

$$\mathbf{C'} = \mathbf{C} + w_i(x'_i - x_i), \text{ mod } m$$

The error goes unnoticed (detection fails) if the change in the check value is zero. That means:

$$w_i(x'_i - x_i) \equiv 0, \text{ mod } m$$

But since the digit change $(x'_i - x_i)$ can be any non-zero number (when we think about all possible digit changes), the only way to be sure detection never fails is if w_i has no common factor with *m*. Mathematical we write gcd(w_i, *m*) = 1

When this is true, we say w_i is invertible modulo *m*.

Transposition errors (swapping two neighboring digits):

Now suppose we swap the digits at positions i and i+1.

The old part of the sum was:

$$w_i x_i + w_{i+1} x_{i+1}$$

The new part becomes:

$$w_i x_{i+1} + w_{i+1} x_i$$

So the total change in the check value is:

$$(w_i x_{i+1} + w_{i+1} x_i) - (w_i x_i + w_{i+1} x_{i+1})$$

This simplifies to:

$$(w_{i+1} - w_i)(x_i - x_{i+1})$$

So the new check value changes by this amount, taken modulo *m*.

The swap goes undetected if this change is zero, meaning:

$$(w_{i+1} - w_i)(x_i - x_{i+1}) \equiv 0, mod\ m$$

even though x_i and x_{i+1} are different (so $x_i - x_{i+1}$ is not zero).

To avoid this, we must choose *m* and all the weights w_i carefully so that this product never becomes a multiple of *m* unless the digits are the same. This gives us specific rules for choosing m and the weights.

Year 9; July, 2026; Vol. 8; Issue 2; PP. (1-12) DOI: https://doi.org/10.3126/amcjd.v8i2.98370

Some Definitions and Notations

We recall some standard definitions (Burton, 2007; Shoup, 2012).

Definition 1: Let $\mathbb{Z}$ be the set of integers. Then any $x, y \in \mathbb{Z}$, and n be fixed positive integer, we say that x is congruent to y modulo n. Symbolically in Gaussian form; $x \equiv y$, mod n if the difference x-y is divisible by n, i.e., $n \mid x$-y. Thus:

$x \equiv y$, mod $n \Rightarrow n \mid x$-y= k (say) $\Rightarrow x$= k $n + y \Rightarrow n \mid x$ and $n \mid y$ gives the same remainder. ♣

If a number n divides the difference of the numbers x and y, then x and y are said to be congruent with respect to n ; if not, they are said to be incongruent. Let 1 8 ≡ 8, mod 5 ∵ 5| 18 leaves remainder 3 and 5| 8 leaves remainder 3. But 18 ≢ 7, mod 5 ∵ 5| 18 leaves remainder 3 and 5| 7 leaves remainder 2. So 18 and 7 are not congruent, i.e., they are incongruent.

Definition 2: (Weighted sum check digit) Given digits $x_1, \cdots, x_k$ and weights $w_1, \cdots, w_k$,

the check digit c is computed as c ≡ $\sum_{i=1}^{k} w_i x_i$, mod m,

where m is the modulus. The full number is $x_1 \cdots x_k$c.

Definition 3: (Error detection) . A scheme detects a single-digit error if changing any

single x_i to $x_i' \neq x_i$ changes the computed check digit. It detects an adjacent transposition if swapping x_i and x_{i+1} ($x_i \neq x_{i+1}$) changes the check digit (Gallian, 2021; Menezes et al., 2018).

Definition 4: Residue Class modulo n is defined by $[x]_n = \{y: y = x + k\,n, k \in \mathbb{Z}\} = \{y: y \equiv x, \text{mod } n\}$

There are n distinct residue classes: [0], [1],⋯,[n-1]. For example residue classes modulo 3 are : $[0]_3$ ={⋯ , -6, -3, 0, 3, 6,⋯ }, $[1]_3$ = { ⋯ ,-7, -4 ,1, 4, 7, ⋯ }, and $[2]_3$ ={ ⋯ ,-8, -5, 2, 5, 8,⋯ }etc. Residue classes are in general parametrized by the remainders in the division algorithm, and there can't be more of them than the number one mods out with (Burton, 2007; Lange, 2002).

Remark 1: We can equip $\mathbb{Z}n$ with binary operations defining addition and multiplication in a natural way as follows: for $x; y \in \mathbb{Z}$, we define

1. $[x] + [y] = [x + y]$;
2. $[x] \cdot [y] = [x \cdot y]$: Observe that for all $x; y; z \in \mathbb{Z}$, we have
3. $[x] + [y] = [z] \Rightarrow x + y \equiv z$, mod n;and
4. $[x] \cdot [y] = [z] \Rightarrow x \cdot y \equiv z$, mod n;

Definition 5 : (Shoup, 2012) Let $a; n \in \mathbb{Z}$ with $n > 0$. We say that $z \in \mathbb{Z}$ is a multiplicative inverse of a modulo n if $az \equiv 1$ (mod n). But it is known that a has a multiplicative inverse modulo n if and only if gcd ($a; n$) = 1.

Some Properties of Congruence Modulo n (Dummit & Foote, 2004; Hungerford, 2013)

1. $x \equiv y$, mod n *is* an equivalent relation.
2. It partitions $\mathbb{Z}$ into equivalence class(residue classes).

3. It arises from Ideal
4. It creates a quotient Ring
5. It fits into the homomorphism/isomorphism theorem frame work.

Results and Discussion

Theorem 1 If $x \equiv y$, mod n, and $u \equiv v$, mod n, then $x+y \equiv u+v$, mod n.

Theorem 2 If $x \equiv y$, mod n, and $u \equiv v$, mod n, then $xy \equiv uv$, mod n

Theorem 3 Two integers are congruent modulo if and only if they belong to the same residue class, i.e,.

$x \equiv y$, mod $n \Leftrightarrow [x]=[y]$.

Theorem 4 (Tuset, 2024) *If $zx \equiv zy$, mod n for relatively prime z and n, then $x \equiv y$, mod n.*

Theorem 5 *If p is a prime number that does not divide an integer x , then $x^{p-1} \equiv 1$, mod p.*

Theorem 1 and Theorem 2 are used as follows

1. Any positive integer is congruent to the sum of its digits, modulo3. This result obtained on the following fact; $10^n \equiv 1$, mod 3 for any integer n, this fact is established as

   $10 \equiv 1$,mod 3

   $10^n \equiv 1$,mod 3, by Theorem 4.2 .

   Consider now the number 3456, which may be written as

   $3456= 3\cdot 10^3 +4\cdot 10^2+5\cdot 10^1+6$, since each power of 10 is congruent to 1, modulo3, then from Theorem 1 and Theorem 2 enable us to write

   $3456 \equiv 3+4+5+6$, mod 3.

   The result illustrated above for 3456 can be demonstrated in the same vein for any +ve integer. A +ve integer is, therefore, divisible by 3 if and only if the sum of its digit is divisible by 3. This example is generalized as follows:

   Let $n \in \mathbb{Z}_+$. Then $10^{2n} \equiv 1$, mod11, $10^{2n-1} \equiv 10$, mod 11 use the result to show that any number of the form da, cbb, cad where a, b, c, d are digits divisible by 11.

Solution: Since $10^{2n} \equiv 1$, mod11, $10^{2n-1} \equiv 10$, mod 11$\Rightarrow 10^{2n}/10 \equiv 1/10$, mod 11

$\equiv -1$, mod 11

$\Rightarrow 10^{2n-1} \equiv -1$, mod 11

$\therefore$ dacbbcad $= d\times 10^7 + a\times 10^6 + c\times 10^5 + b\times 10^4 + b\times 10^3 + c\times 10^2 + a\times 10^1 + d\times 10^0$

$\equiv (d\times (-1) + a\times 1 + c\times (-1) + b\times 1 + b\times (-) + c\times 1 + a\times (-1) + d\times 1)$, mod 11

$\equiv 0$, mod 11

⇒dacbbcad ≡ 0, mod 11 [∵ $10^7=10^{2.3-1}$ ≡-1 mod 11, $10^6=10^{2.3}$ ≡1 mod 11 etc from given conditions] A number is divisible by 11 if the sum of its digit is divisible by 11. Similar case will be seen for any number,

2. It helps determining the day of the week for any date in the future since weeks repeat every 7 days.

3. It is used to determine the time, thinking it as 12 hours clock.

Example 1. Show that 1,000,000≡ 1, mod 7.

Solution:

We have 10≡ 3, mod 7.

10^2 ≡ 9≡ 2,mod 7.

10^6 ≡ 2.2.2.≡ 8≡ 1 mod 7

Using the result of Example 2, we determine the day of the week for any date in the future since weeks repeat every 7 days.

Some basic modular arithmetic and time and clock problems

Problem 1: What day of the week will it be 1,000,000 days from the day on which you are reading this problem?

Solution: Suppose today is 14-02-2026; Saturday is the day on which we are reading this problem. We need to find the day of the week. That is 1,000,000 days from today. Days of the week repeat every 7 days. Thus our problem is to find 1,000,000≡ 1, mod 7, and see what day that is from today (∵ a date falls on mod 7).

Step 1. 7| 10 leaves remainder 3, so

10≡ 3, mod 7

⇒ 10^2 ≡ 3^2 = 9 ≡2; mod 7(by Th2.)

⇒ 10^3≡ 2× 3 ≡ 6 ; mod 7

⇒ 10^4≡ 6× 3 ≡ 18 ≡ 4; mod 7

10^5≡ 4× 3 ≡ 12 ≡ 5; mod 7

10^6≡ 5× 3 ≡ 15 ≡ 1; mod 7

∵ 10^6≡ 1, mod 7, the pattern repeats every 6 power of 10 in modulo 7

Step 2. Calendar Calculation: Starting day; Saturday (call it day 0)

Add 1 day → Sunday (1 day).

So **1,000,000 days from Saturday is a Sunday.**

Problem 1:

What modulus is used in dealing with numbers associated with the days of the week? Show that 365≡ 1, mod 7. If January falls on Wednesday in a year which is not a leap year, on which day of the week will January 1 of the following year fall?

Solution

We use modulus 7. Since normal year has 365 days (not included leap year), 365= 7× 52 +1,

since remainder is 1, we write in modular form; 365≡ 1, mod 7. This means that a normal year has 1 additional day beyond complete weeks. So the calendar shifts forward one day in the next year. Thus if January 1 is Wednesday, after 365 days one extra day is added on the previous January 1, which is Wednesday. Therefore, the day of the week of the following year will be Wednesday +1 = Thursday (for non-leap year).

Remark1. If it a leap year, then 365≡ 2, mod 7 and the day would shift by two days instead ,i.e., the following January is Wednesday +2 = Friday.

Problem 2:

The 24-hour time notation (like 15:00) is common in various countries. The conversion to a 12-hour system is done by taking the hour value modulo 12. This system applies to tell the time: for example in 12 hours clock, 15:00 P.M. is same as 3:00 P.M.

Since in modular form we write

15 ≡ 3, mod 12 ∵ 12| 15-3 ⇒ 15 mod 12=3, so 15:00 is also told 3:00 PM. In particular: 0:00 (midnight) and 12:00 (noon) are both 12 in 12-hour clocks, though mathematically

12≡0, mod 12. We simply label 0 as 12 for convenience.

How to know the usual time:

To know what time it will be after a certain number of hours, the hours is added and then reduced mod12. Example: It's 9:00 now. What time will it be in 5 hours? 9+5=14, and 14 mod 12 =2, so it will be 2:00. Similarly, if you add 24 hours (a full day), the hour stays the same because 24 ≡ 0, mod 12.

Problem 3:

If $x \equiv 2$, mod 5 and $0 < x < 30$. then write the set of values of x.

Solution:

If $x \equiv 2$, mod 5⇒ 5| x-2= k ⇒ $x = 5k + 2$ for some integer k, so here $0 < 5k+2 < 30$.

Solve: $5k+2 > 0$ ⇒$5k > -2$ ⇒ $k > -0.4$, so $k \geq 0$ since it is integer. Also $5k+2 < 30$ ⇒$5k < 28$ ⇒$k < 5.6$, so $k \leq 5$. So the possible values of k = 0, 1, 2, 3, 4, 5.

Then x = 2,7,12,17,22,27. So the set is {2, 7, 12, 17, 22, 27}. Thus values of x satisfying

$x \equiv 2 (\text{mod} 5)$ and $0 < x < 30$ are: {2, 7, 12, 17, 22, 27}.

Solving linear congruencies (Lang, 2002):

We solve the linear congruence equations like :

$$ax \equiv b, \bmod n \cdots \quad (1)$$

To solve equation (1), first check whether its solution exists or not. If the greatest common divisor (gcd) of a and n divides b, i.e., (a, n)|b then its solution exists otherwise not .

Problem 4 :

Solve the linear congruence equation

$x \equiv 2, \bmod 3$

a=1, n=3, b=2

gcd(1,3)=1. Since 1|2 =2 So its solution **exists**.

General solution is

$$x=2+3k,\ k \in \mathbb{Z}$$

So the solution set is {⋯ , -4, -1, 2, 5, 8,⋯ }. Usually we express it as $x \equiv 2$, mod 3. A linear congruence equation; $2x \equiv 1$, mod 6 has no solution because the greatest common divisor of 2 and 6 is 2 but 2 ∤1. For a linear congruence $ax \equiv b$, mod n to have a solution, we need gcd (a, n)|b. Here, that condition is not met, so the equation is inconsistent. Note: An even number can never be congruent to an odd number modulo 6.

New Theorems and Proofs:

Theorem 6: (Single-digit error detection condition). A weighted sum check digit $c \equiv \sum_i^k w_i x_i$, ***mod m detects all single-digit errors iff gcd(*** w_i, ***m)=1 for every i, m ≥ 2 and weights*** $w_1, \cdots, w_k \in \mathbb{Z}$.

Proof.: ``⇒" Suppose some gcd(w_i, m) = d > 1 for some i. Then $d \mid m$ and $d \mid w_i$. Choose $\delta = d \mid m$ (non-zero modulo m). Then $\delta \not\equiv 0$, *mod* m and $w_i\delta = w_i \cdot (m/d) = (w_i/d).m \equiv 0$, mod m. Choosing $x_i' = x_i + \delta$ (which is a different digit when working with residues modulo m) leaves the check digit unchanged; thus a single-digit error is not detected. Hence ∀ I, gcd(w_i, m)=1 is necessary.

``⇐" If gcd(w_i, m) = 1, then w_i is invertible modulo m. For any non-zero digit change $\delta = x'_i - x_i$, we have $w_i\delta \not\equiv 0$, mod m because w_i invertible and $\delta \neq 0$, so the check digit changes → detection succeeds.

Theorem 7: (Adjacent transposition detection condition). All adjacent transpositions are detected if and only if m is prime and $wi \not\equiv w_{i+1}$, mod m for all i.

Proof. The change in the check sum after swapping x_i and x_{i+1} is $\Delta=(w_{i+1}-w_i)(x_i-x_{i+1})$. Detection fails iff there exist digits $xi \neq xi+1$ such that $\Delta \equiv 0$ (mod m).

Case 1 If m is composite: Let d be a proper divisor of m. Choose digits such that $x_i - x_{i+1} = m/d$ (non-zero) and suppose $w_{i+1} - w_i$ is a multiple of d. Then Δ is a multiple of m → undetected. Hence composite moduli cannot guarantee detection.

Year 9; July, 2026; Vol. 8; Issue 2; PP. (1-12) DOI: https://doi.org/10.3126/amcjd.v8i2.98370

Case 2 If m is prime, then $\Delta \equiv 0$, mod m implies either $w_{i+1} \equiv w_i$, mod m or $x_i \equiv x_{i+1}$,mod m. Since $x_i \neq x_{i+1}$ and digits are less than m (we assume m > 10 for prime case, but for m = 11, digits 0–9 are all distinct mod 11), the only way to fail is $w_{i+1} \equiv w_i$, mod m. Thus if all adjacent weights are distinct mod m, every transposition changes the sum, i.e., the product is non-zero → detection guaranteed.

Conversely, if m is prime but $w_i \equiv w_{i+1}$**,** then $\Delta \equiv 0$ for any transposition → detection fails. Thus distinctness is necessary.

Corollary 1. For modulus m = 10 (composite), no weight vector can detect all adjacent

transpositions. For modulus m = 11 (prime), choosing distinct weights modulo 11 guarantees detection of all adjacent transpositions.

3.4 Applications (Rosen, 2019; Burton, 2007)

The Concept of Error Detection: In identification systems like the ISBN (for books) and bank routing numbers, a single mistyped digit can lead to misidentification or financial error. To prevent this, a **check digit** is appended to the number. This digit is calculated mathematically from the preceding digits using **modular arithmetic**. When the full number is entered, the system recalculates this check digit; if it does not match the appended one, an error is flagged.

ISBN-10 (Modulo 11)

The International Standard Book Number (ISBN-10) used in many libraries consists of nine digits $x_1 x_2 \ldots x_9$ followed by tenth check digit x_{10} which satisfies

$x_{10} = \sum_{k=1}^{9} k \cdot x_k \, , mod 11 \cdots \cdots (1)$ It is called ISBN check digit condition.

It uses a weighted sum modulo 11 to catch common errors like a single wrong digit or the transposition of two digits. Here, mod 11 is used because a prime number, this system can detect 100% of single-digit errors and transposition errors. If the computed sum modulo 11 equals 10, the check digit is written as X.

$\therefore$ we use $x_{10} \equiv$ that sum, mod 11.

Example 2:

Determine whether each of the ISBNs below is correct:

(a) ISBN "0-07-232569-0"(United States)

(b) ISBN "91-7643-497-5"(Sweden)

(c) ISBN "1-56947-303-10"(England)

Solution :

(a) 0-07-232569-0:
Digits: x_1=0, x_2=0, x_3=7, x_4=2, x_5=3, x_6=2, x_7=5, a_8=6, x_9=9. Using (1) we write

Sum = 1·0+2·0+3·7+4·2+5·3+6·2+7·5+8·6+9·9=2201·0+2·0+3·7+4·2+5·3+6·2+7·5+8·6+9·9=220

$\therefore 220 \equiv 0$, mod11, and $x_{10}(=a_{10})=0 \rightarrow$ **correct**.

(b) 91-7643-497-5:

Digits: $x_1=9, x_2=1, x_3=7, x_4=6, x_5=4, x_6=3, x_7=4, x_8=9, x_9=7$

Sum = $1\cdot9+2\cdot1+3\cdot7+4\cdot6+5\cdot4+6\cdot3+7\cdot4+8\cdot9+9\cdot7=2571\cdot9+2\cdot1+3\cdot7+4\cdot6+5\cdot4+6\cdot3+7\cdot4+8\cdot9+9\cdot7=257$

$\therefore$ $257 \equiv 4$, mod11, but $x_{10}= 5 \rightarrow$ **incorrect**.

(c) **1-56947-303-10**:

Digits: $x_1=1, x_2=5, x_3=6, x_4=9, x_5=4, x_6=7, x_7=3, x_8=0, x_9=3$

Sum = $1\cdot1+2\cdot5+3\cdot6+4\cdot9+5\cdot4+6\cdot7+7\cdot3+8\cdot0+9\cdot3=1751\cdot1+2\cdot5+3\cdot6+4\cdot9+5\cdot4+6\cdot7+7\cdot3+8\cdot0+9\cdot3=175$

$\therefore 175 \equiv 10$, mod11, and $x_{10}=10 \rightarrow$ **correct**.

Remark Here the last part is "10" meaning check digit is 10? Typically in ISBN-10, check digit 10 is written as X. But here they wrote 10, so we treat $a_{10} = 10$.

Bank Identification Numbers (Modulo 10)

Bank identification numbers (often part of routing or account numbers) use a weighted sum mo dulo 10. For eight digits x_1 to x_8, the check digit x_9 is given by:

$$x_9 \equiv 7x_1+3x_2+9x_3+7x_4+3x_5+9x_6+7x_7+3x_8, \text{ mod10} \cdots \cdots \quad (2)$$

For bank IDs, check digit x_9 is the sum of first eight digits with repeating weights 7,3,9 modulo 10, designed to catch single-digit and adjacent transposition errors, i.e, $x_9 \equiv$ that sum mod 10.

Example 2 . Consider the eight digit Bank identification numbers $x_1x_2\cdots x_8$ which is followed by a ninth digit check digit x_9 chosen to satisfy equation (2):

(a) Obtain the check digit that should be appeared to the numbers 55382006 and 81372439.

(b) The bank check identification numbers $237x_418538$ has an illegible fourth digit. Determine the value of obscured digit?

Solution

Part (a) For an eight-digit bank IDs, $x_1x_2\ldots x_8$, the check digit x_9 is given by equation(2)

For 55382006:

Digits: $x_1=5, x_2=5, x_3=3, x_4=8, x_5=2, x_6=0, x_7=0, x_8=6$

Weighted sum $= 7\cdot 5+3\cdot 5+9\cdot3+7\cdot 8 +3\cdot 2 +9\cdot 0+7\cdot0+3\cdot 6$

$=35+15+27+56+6+0+0+18=157$

$\therefore$ $157 \equiv 7$,mod10) $\rightarrow$ check digit **7**.

For 81372439:

Digits: $x_1=8, x_2=1, x_3=3, x_4=7, x_5=2, x_6=4, x_7=3, x_8=9$

Weighted sum = $7\cdot8+3\cdot1+9\cdot3+ 7\cdot 7+ 3\cdot 2+ 9\cdot 4+ 7\cdot 3+3\cdot 9= 3\cdot9=273\cdot9=27$

$= 56+3+27+49+6+36+21+27$

$= 225$

$\therefore$ $225 \equiv 5, \text{mod}10 \rightarrow$check digit **5**.

Part (b) The number is ``237x418538"(with x_4 unknown, $x_9 = 8$):

$x_1=2$, $x_2=3$, $x_3=7$, $x_4=?$, $x_5=1$, $x_6 = 8$, $x_7=5$, $x_8=3$, $x_9=8$.

The check digit condition gives $8 \equiv 7 \cdot 2+3 \cdot 3+9 \cdot 7+7x_4+3 \cdot 1+9 \cdot 8+7 \cdot 5+3 \cdot 3, \text{mod}10$.

Sum = 14+9+63+3+72+35+9=20514+9+63+3+72+35+9=205

Thus $205+7x_4 \equiv 8, \text{mod}10$.

Since $205 \equiv 5, \text{mod}10$, we have $5+7x_4 \equiv 8 \Rightarrow 7x_4 \equiv 3, \text{mod}10$.

Checking digits 0–9: $7x_4$, mod 10 yields 3 when $x_4 = 9$.

Hence the obscured digit is **9**.

Calculation: We need $(205 + 7a_4) \bmod 10 = 8$. Compute $205 \bmod 10 = 5$,

so $(5 + 7x_4), \text{mod } 10 = 8 \equiv 7x_4, \text{mod } 10 = 3$ (since 8-5=3). So we need $7x_4 \equiv 3 \bmod 10$. Solve for x_4 digit 0-9. $7x_4 \bmod 10$:

7*0=0, 7*1=7, 7*2=14→4, 7*3=21→1, 7*4=28→8, 7*5=35→5, 7*6=42→2, 7*7=49→9, 7*8=56→6, 7*9=63→3. So $a_4=9$ gives 63 mod10=3. Thus $a_4 = 9$.

So answer: (a) 7 and 5; (b) 9.

Number '237x418538' (with x4 unknown, x9 = 8):

$8 \equiv 7 \cdot 2 + 3 \cdot 3 + 9 \cdot 7 + 7x4 + 3 \cdot 1 + 9 \cdot 8 + 7 \cdot 5 + 3 \cdot 3 \text{ (mod 10)}$.

Sum of known terms = $14 + 9 + 63 + 3 + 72 + 35 + 9 = 205 \equiv 5 \text{ (mod 10)}$.

Thus $5 + 7x4 \equiv 8 \text{ (mod 10)} \Rightarrow 7x4 \equiv 3 \text{ (mod 10)}$. Testing digits, x4 = 9 gives

$7 \cdot 9 = 63 \equiv 3$. Hence the obscured digit is 9.

Future Line of Research (Bernstein & Lange, 2017).

- Congruence modulo n is classical, but there are still many directions remain open for new work. Congruence generalizes to an arbitrary ring R: $a \equiv b(\text{mod}, I)$ where I is an ideal in general ring R. This congruence studies Quotient Ring R/I, ring homeomorphisms and ring structure theory. Potential research topics include:
- Generalizing check-digit algorithms to other moduli and algebraic structures.
- Applications of modular arithmetic in post-quantum cryptography.
- Congruence in non-commutative rings and its computational aspects.

Conclusion

The notion of congruence modulo *n* in number theory, along with its modern algebraic formulation and foundational implications in comparison to ancient modular arithmetic concepts were introduced and the primary focused practical numerical problems were solved, including linear congruences, clock arithmetic, and error-detection systems such as ISBN (modulo 11) and bank identification numbers (modulo 10).

Acknowledgment

The author thanks the mentors for their help, support, and useful suggestions. Also, thanks to the referee of this paper for the valuable comments.

References

Bernstein, D.J., & Lange, T. (2017). Post-quantum cryptography. *Nature*, 549(7671), 188–194. https://doi.org/10.1038/nature23461

Booth, W.C., Colomb, G.G., & Williams, J.M. (2016). *The craft of research* (4th ed.). University of Chicago Press.

Burton, D.M. (2007). *Elementary number theory* (6th ed.). McGraw-Hill Higher Education.

Dummit, D.S., & Foote, R.M. (2004). *Abstract algebra* (3rd ed.). Wiley.

[5]Gallian, J.A. (2021). *Contemporary abstract algebra* (10th ed.). Cengage.

Harel, G. (2013). What is mathematics? A pedagogical answer to a philosophical question. In R.B. Gold & R. Simons (Eds.), *Proof and other dilemmas: Mathematics and philosophy* (pp. 265–290). MAA Press.

Hungerford, T.W. (2013). *Abstract algebra: An introduction* (3rd ed.). Cengage.

Lang, S. (2002). *Algebra* (Revised 3rd ed.). Springer.

Menezes, A.J., van Oorschot, P.C., & Vanstone, S.A. (2018). *Handbook of applied cryptography*. CRC Press.

Rosen, K.H. (2019). *Elementary number theory and its applications* (6th ed.). Pearson.

Shoup, V. (2012). A computational Introduction to Number Theory and Algebra. *Cambridge University Press*, DOI: 10.1017/CBO 9781139165464.

Swales, J.M. (2014). *Genre analysis: English in academic and research settings*. Cambridge University Press.

Thomas, W.J. and Rober, A.B. (2020). *Abstract Algebra: Theory and Applications.* (Judson), mail to:info@LibreTexts.org, https://math.libretexts.org/@go/page/81017

Tuset, L. (2024). *Abstract Algebra via Numbers*. Springer Cham., https://DOI:10.1007/978-3-03/2-74623-9.